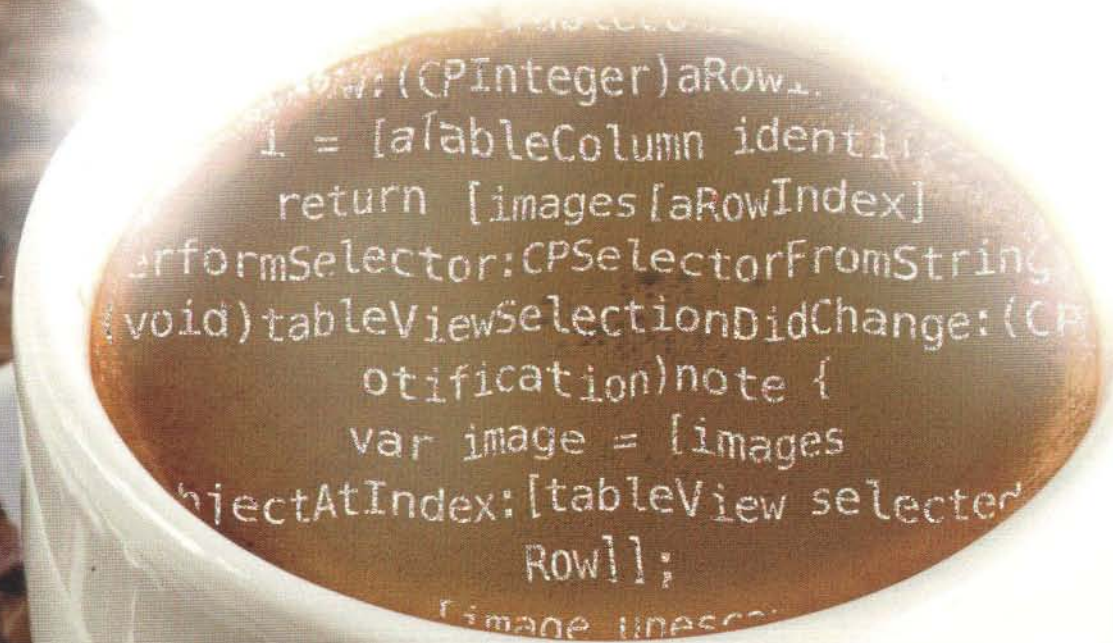
Developing Enterprise-Friendly Software for Fun and Profit

MacTech Magazine
May 2011 • Issue 27.05

MACTECH®

The Journal of Apple Technology

Creating Web Applications with Cappucino



Mac in the Shell:
Adding a Cocoa GUI to a MacRuby application

Developer to Developer:
No Pane, No Gain

MACTECH.COM

$8.95 US, $12.95 Canada

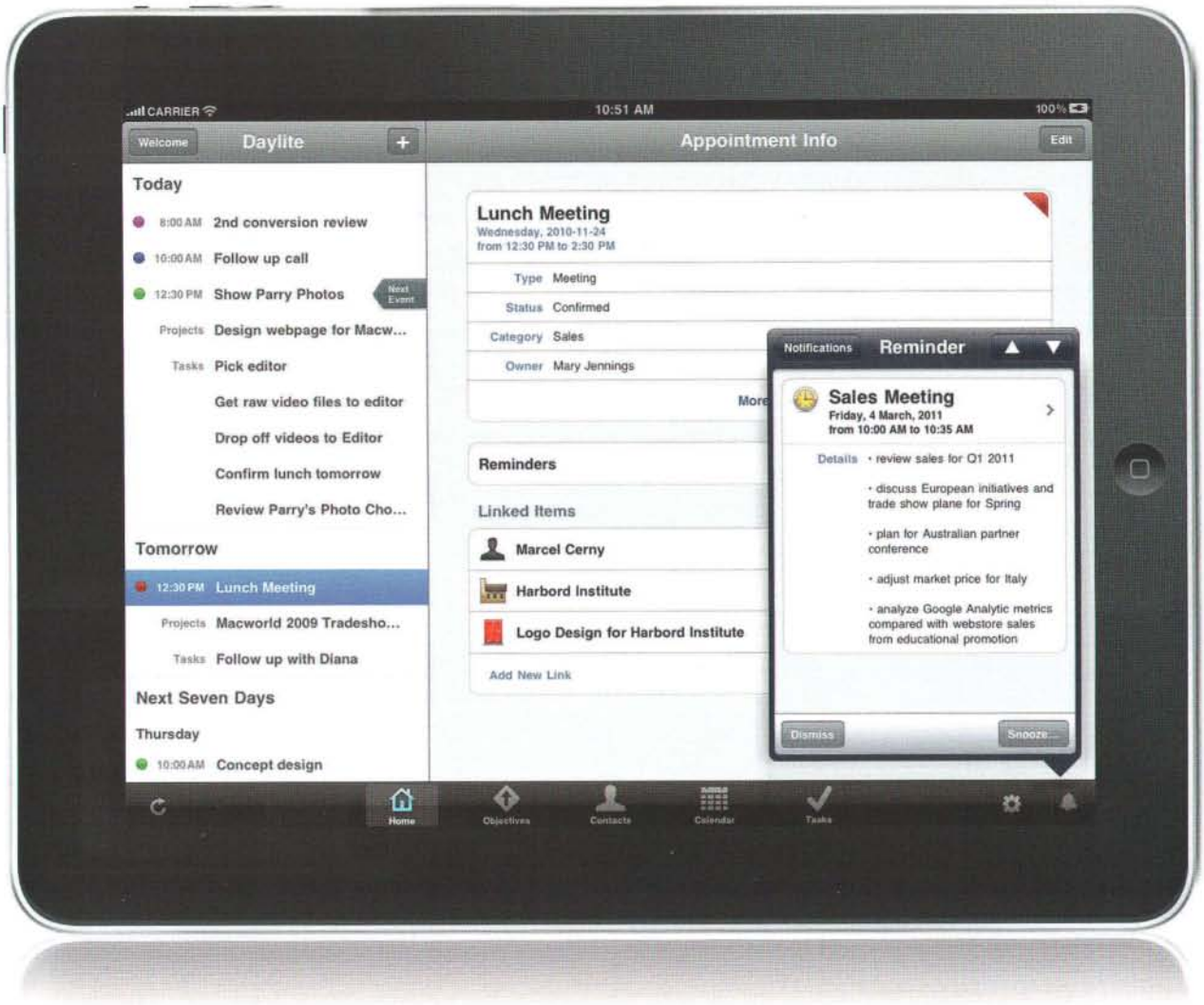
05

0 32128 74887 8

ISSN 1067-8360 Printed in U.S.A.

Your business at your fingertips.

With Daylite for iPad you'll have access to your critical business data no matter where you are, letting you make important business decisions even when you and your employees are on different sides of the world.

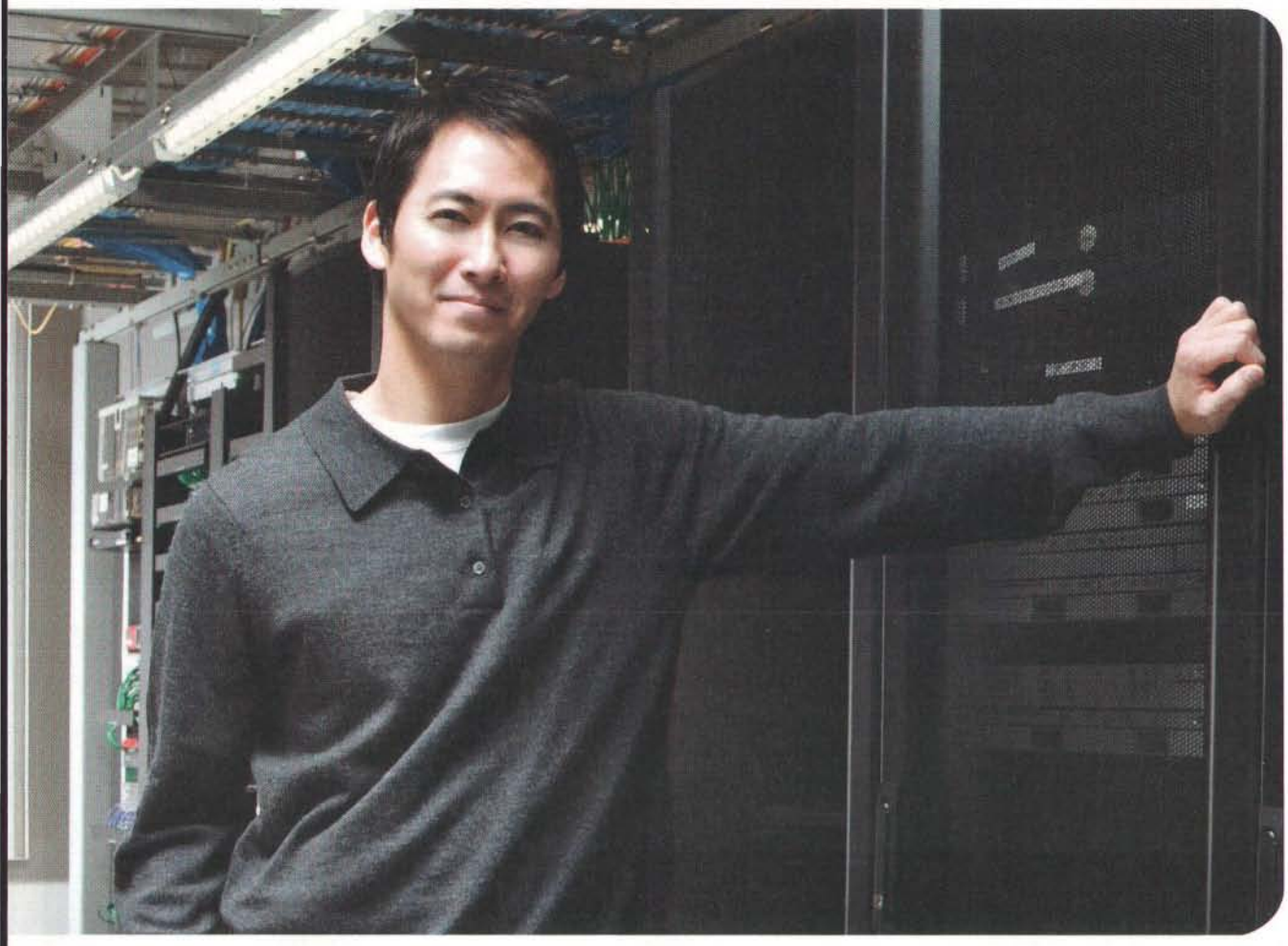


Daylite

Visit www.marketcircle.com/daylite for a free 30-day trial.

The idea:
Turn surfboards into art
Serena Mitnik-Miller and Mason St. Peter of custom surfboard collective Two Birds Fly are living proof that when you're passionate about an idea, and you free it, beautiful things happen. Whether they're managing orders or creating promotional materials, working in a trusted productivity suite helps them take care of business and make sure their big idea takes flight.
Free the ideas.
Office:mac 2011
HOME & STUDENT
Office:mac 2011
HOME & BUSINESS
Introducing Office:mac 2011
Transcend platforms and free your ideas with Microsoft® Office for Mac 2011.
Learn more at officeformac.com/freetheideas

Parallels



Parallels Server for Mac 4.0

Fully utilize your Xserve, Mac Pro and Mac mini investment.

Consolidate and run any server software you choose.

- Full scale hypervisor solution with bare metal architecture
- Hardware-ready for seamless integration into existing IT infrastructures
- Built-in VM and multiple server management and maintenance tools
- Point and click migration of VM's between physical servers
- Integrated snapshots and incremental and full backups included

"Our district has already saved more than $50,000 because Parallels Server for Mac eliminates the need for multiple servers. We anticipate more savings as well."

—Richard Bowler
Director of Technology
Hardin School District in Montana

To learn more, visit www.parallels.com/products/server/mac

Parallels

My Mac does do Windows.

Parallels Desktop® 6 for Mac.
Simply Faster, Smarter and More Powerful Than Ever!

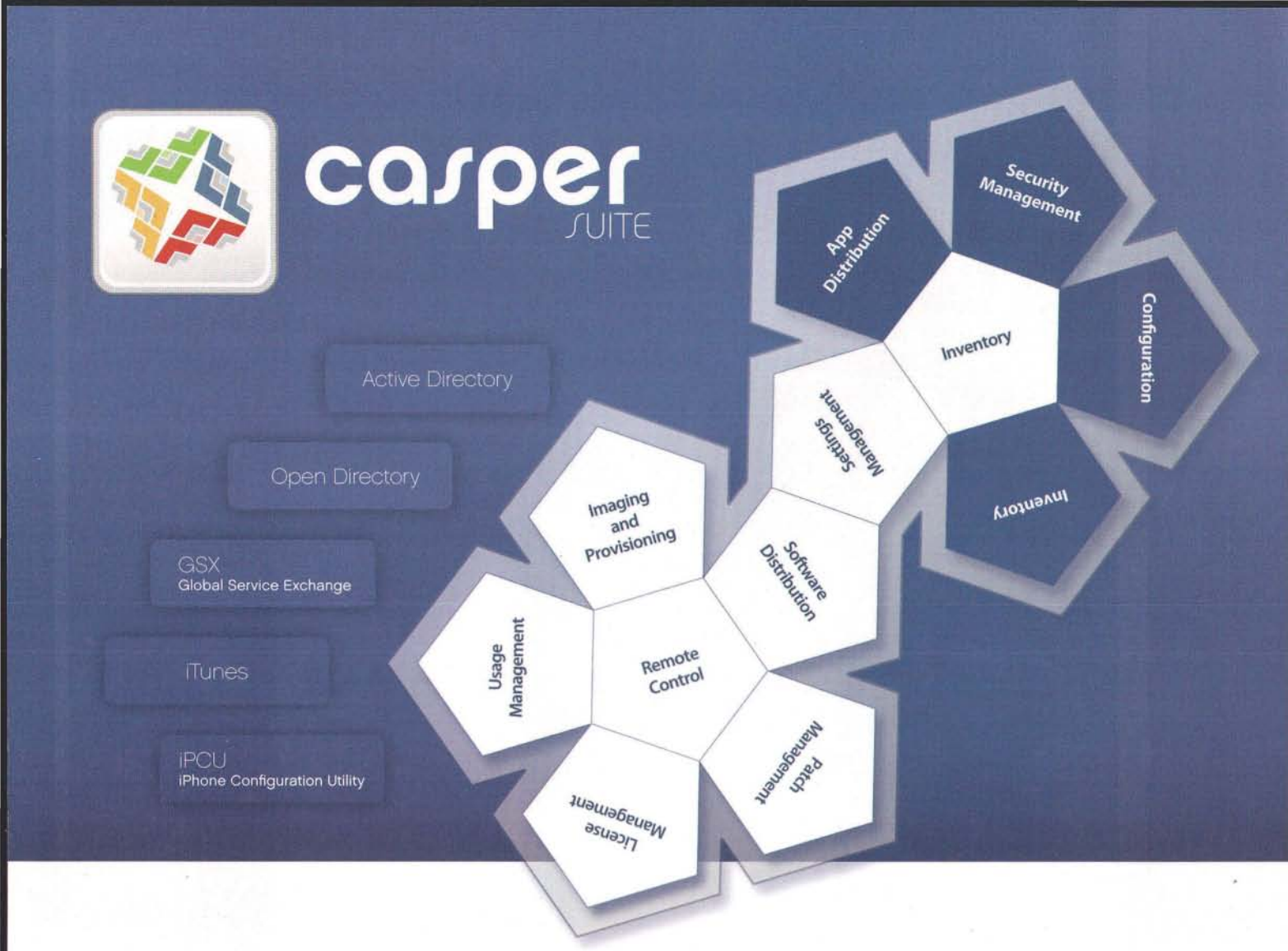
- **New!** Get full control of your virtual machine with our all-new Parallels Mobile app for iPhone/iPad
- **New!** Take advantage of all of the capabilities of your 64-bit Mac and enjoy our fastest virtual machine performance yet
- **Enhanced!** Experience brilliant graphics capabilities in Windows applications whether you're a gamer, architect, designer or engineer
- **New!** Immerse yourself in your favorite games, music and videos with rich 5.1 Surround Sound

For business or for home, Parallels is the #1 choice of customers worldwide*

To learn more, visit www.parallels.com/desktop today.

Award winning reviews!

*No. 1 claim based on NPD Techworld Reports for the period 8/1/06 - 09/30/10



Casper Suite 8

Comprehensive iOS Management capabilities for iPad, iPhone and iPod touch devices.

Administrators can now manage iOS devices using the same console they use to manage their Macs, benefitting from deep integration with the Apple ecosystem while leveraging their existing technology.

iOS

Comprehensive iOS Management capabilities

Self Service console for over-the-air (OTA) distribution of in-house and App Store apps

Unification and extensibility: helping the enterprise succeed with the Apple platform

Weekly Demo

Learn more about Casper Suite 8

www.jamfsoftware.com/demo

http://www.jamfsoftware.com

JAMF software

HELPING THE ENTERPRISE SUCCEED WITH THE APPLE PLATFORM

TABLE OF CONTENTS

Mac in the Shell
Ruby and the GUI
Adding a real Cocoa GUI to a MacRuby application
by Edward Marczak 8

Consultant Cowboy
Pricing (Part 2)
More thoughts on pricing
by Ryan Wilcox 28

MacEnterprise
Developing Enterprise-Friendly Software for Fun and Profit
by Greg Neagle, MacEnterprise.org 38

CoreSec
Secure Shell (SSH)-A Hidden Gem, Part 2
SSH is a powerful tool with many uses. Let's explore this staple of OS X
By Michele (Mike) Hjörleifsson 48

Meet Cappucino
Use your Cocoa skills to build stunning web applications
by Johannes Fahrenkrug 52

Developer to Developer
No Pane, No Gain
Building a Preference Pane plug-in to integrate into System Preferences
by Boisy G. Pitre 62

The EDA Showcase
Scripting Automations in Windows for Mac OS X Administrators
by Charles Edge 66

The SQLite DB and iOS
How to use SQLite on your iPhone and iPad
by Mihalis Tsoukalos 80

Adaptive Laptop Management with crankd
by Gary Larizza 88

Kool Tools
VMware View Client for iPad
Windows on iOS—to go!
by Dennis Sellers 92

The MacTech Spotlight
Justin Williams, Second Gear 96

From the Editor

It's WWDC time, and at this point, since the event has sold out, you're either attending or not. Of course, you *should* be attending because conferences are for you. Really. Developers, this is a no-brainer: there's so much to learn, particularly this year with a new OS release on the horizon. iOS developers, you're not excluded, either. IT people: if this year's WWDC didn't interest you, you're a bit lost. Whaaaa?

IT staff have one of the toughest jobs: know everything. No one asks developers to set up a network. No one asks developers to manage a fleet of hundreds of machines. No one asks a developer to reset a user account or correct permissions on a home directory. But that—and more—is IT. You might think the 'and more' part is setting up printers or training users. Sure, that's a part of it. My issue is that the 'and more' needs to include development. Again: whaaaa?

When one person needs to configure their machine for VPN access, you can walk them through it over the phone or visit them. When 1,000 machines need to be configured for VPN access, you had better figure out a way to automate the task. This involves some amount of development time. When you'd like to trigger actions that take place for your fleet of machines when they come onto your corporate LAN, that's going to take some development time. If you'd like to ease the use of setting some machine preferences with the use of a custom Preference Pane, that's going to take some development time. IT people, you need to be developers, too.

Now, IT people don't need to master every facet of development. You don't need to be Mark Dalrymple or Aaron Hillegass. But a modicum of Ruby? A smidge of Python? Basic Objective-C? Yes! Understanding basic development will *also* help you understand crash logs and aid you generally when Things Go Wrong. If you're still afraid of a shell prompt, get over it: Mac OS X has been out for *eleven years*—there's just no excuse any longer.

No matter your discipline, conferences are for you because that's where you meet the *people* that are dealing with this technology every day. The Mac community has the gift of many, many conferences. If you're a web developer, there are HTML and Rails conferences. If you're a Mac developer, there is WWDC, of course, but also many independent conferences through the year, including MacTech Conference. IT people *seemingly* have less choices, however, that's a fallacy. IT people should be attending IT conferences along with developer conferences. Not only do you have something to learn there, but you also have something to offer: a technical user of the developer's products. How you deploy products to end-users is a worthwhile conversation to have with a developer. That's along with the resources used on a machine, or bandwidth used by an application: these are all things that developers assume are OK until they hear it from someone actually in the trenches. MacTech Conference offers developer and IT tracks, along with time for all disciplines to meet.

This month's cover story introduces Cappuccino, a familiar way for Objective-C developers to create web sites, instead, using Objective-J. Think about creating web pages visually with Interface Builder making connections, but coding in Objective-J (for JavaScript). It's very, very cool, and you can let Johannes Fahrenkrug get you started.

This month's Mac in the Shell rounds out a way for Sys Admins to create a GUI-based app that makes it easy for end-users to collect specific logs and compress them for your use.

First-time MacTech author Gary Larizza eases you into the power of crankd, an automation tool that can perform tasks based on system events. Crankd, by the way, is a perfect example of the SysAdmin/developer benefit. As a part of the PyMacAdmin suite of tools, crankd is a Python-based application created by Sys Admins, for Sys Admins.

These, and all of articles in this month's issue showcase the awesome Mac community from nearly every angle. Speaking of showcasing the community, the MacTech Spotlight each month focuses on one member of this community. This month, we hear from Justin Williams who is owner/lead developer/crew chief of Second Gear Software. Second Gear makes some clever, focused applications (for both Mac OS X and iOS). Check out this month's MacTech Spotlight for some advice from Justin.

If you're at WWDC, please, contact us and say hello! (That's @marczak or @mactech on Twitter). If not, find your conference! (and let us know what it is!)

Ed Marczak,
Executive Editor

Communicate With Us

Department E-Mails

Orders, Circulation, & Customer Service
cust_service@mactech.com

Press Releases
press_releases@mactech.com

Ad Sales
adsales@mactech.com

Editorial
editorial@mactech.com
(Authors only, no pr)

Accounting
accounting@mactech.com

Marketing
marketing@mactech.com

General
info@mactech.com

Web Site
http://www.mactech.com

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail?**

If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

If you would like a subscription or need customer service, feel free to contact MacTech Magazine Customer Service at 877-MACTECH

We love to hear from you! Please feel free to contact us with any suggestions or questions at any time.

Write to letters@mactech.com or editorial@mactech.com as appropriate.

MACTECH®

The Journal of Macintosh Technology

A publication of XPLAIN CORPORATION

The Magazine Staff

Publisher & Editor-in-Chief: Neil Ticktin
Executive Editor: Edward R. Marczak
Business Editor: Andrea Sniderman
Ad Director: Bart Allan
Production: David Allen
News: Dennis Sellers
Podcast Producer: Josh Long
drupalmaster: Erik Peterson

Xplain Corporation Senior Staff

Chief Executive Officer: Neil Ticktin
President: Andrea J. Sniderman
Accounting: Marcie Moriarty
Customer Relations: Susan Pomrantz

Columnists

Mac In The Shell: by Ed Marczak
Swaine Manor: by Michael Swaine
KoolTools/Geek Guides: by Dennis Sellers
MacEnterprise: by Philip Rinehart and Greg Neagle
Greg's Bite: by Greg Mills

Regular Contributors

José R.C. Cruz, Michael Göbel, Michele Hjörleifsson, Mihalis Tsoukalos
Oliver Pospisil, Rich Morin, William Smith

Canada Post: Publications Mail Agreement #41513541
Canada Returns to be sent to: Bleuchip International, P.O. Box 25542, London, ON N6C 6B2

MacTech Magazine (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 705 Lakefield Road, Suite I, Westlake Village, CA 91361. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are $47.00 per year. Canadian subscriptions are $59.00 per year. All other international subscriptions are $97.00 per year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

POSTMASTER: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

Opinions expressed are not necessarily the views of MacTech Magazine or Xplain Corporation. All contents are Copyright 1984-2011 by Xplain Corporation. All rights reserved. MacTech is a registered trademarks of Xplain Corporation. MacNews, Xplain, Explain It, MacDev-1, THINK Reference, NetProfessional, NetProLive, Apple Expo, MacTech Central and the MacTutorMan are trademarks or service marks of Xplain Corporation. Sprocket is a registered trademark of eSprocket Corporation. Other trademarks and copyrights appearing in this printing or software remain the property of their respective holders.

by Edward Marczak

Ruby and the GUI

Adding a real Cocoa GUI to a MacRuby application

Introduction

The last few columns have been all about Ruby. From the very basics of the Ruby language to the coupling of Cocoa with MacRuby, we've covered it. This month, we'll use Xcode to create a full Cocoa application. Xcode will let us create both the code and the GUI and connect them together. If you've never used Xcode before, don't be scared off. It's really painless, and the article will be a visual guide.

The Project

The project itself will be a log collector. Oftentimes, an end user will have an issue and you just want them to gather up all of their logs and send them to you. This application will help you with that by defining a list of log files you'd like put on a disk image (.dmg file), which will be stored on a user's desktop, making it easy to retrieve.

Like other projects in this series, the application will work, but be less than perfect; we'll improve it as we go along and in future articles. The most important thing to illustrate is how Xcode allows the GUI and the Ruby code to interact.

To get started, you'll need a Macintosh with Xcode. I'm using Xcode 4 in this article, although you should be able to translate it to Xcode 3. Frankly, Xcode 4 shows the direction Apple is moving in and as a developer, you need to keep up with. Also, Xcode 4—while fine for our example—is less than perfect, and in those cases, bugs need to be reported to Apple. There's no better way to be a model digital citizen than finding and reporting bugs. Find a bug? Report it at http://bugreporter.apple.com.

If you already have Xcode 4 (and it's installed), great. If you need it, either you're part of the developer program and can just download it from http://developer.apple.com, or, you can purchase it for $4.99 from the Mac App Store. In either case, it's a 4GB+ download, so plan for that time and a roughly 10-minute install.

Additionally, don't forget that MacRuby is a separate install. As of this writing, version 0.10 (zero-dot-ten) is current, and can be downloaded from http://macruby.org. Installing the downloaded package will install the MacRuby framework, examples and Xcode templates for MacRuby development.

The Beginning

Before writing any code or creating a user interface, we need to create a new project in Xcode. Launch Xcode and you'll see an introduction screen. Double-click on "Create a new Xcode project" from the choices on the left (see Figure 1).

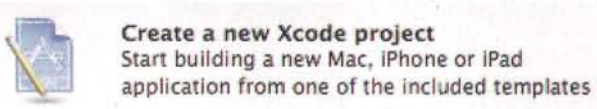


Figure 1 – Create a new project.

Once chosen, you're immediately asked to "Choose a template for your new project." If you've installed the latest MacRuby release properly, you'll see a choice for "MacRuby Application" (see Figure 2). Choose "MacRuby Application" and click the 'Next' button.

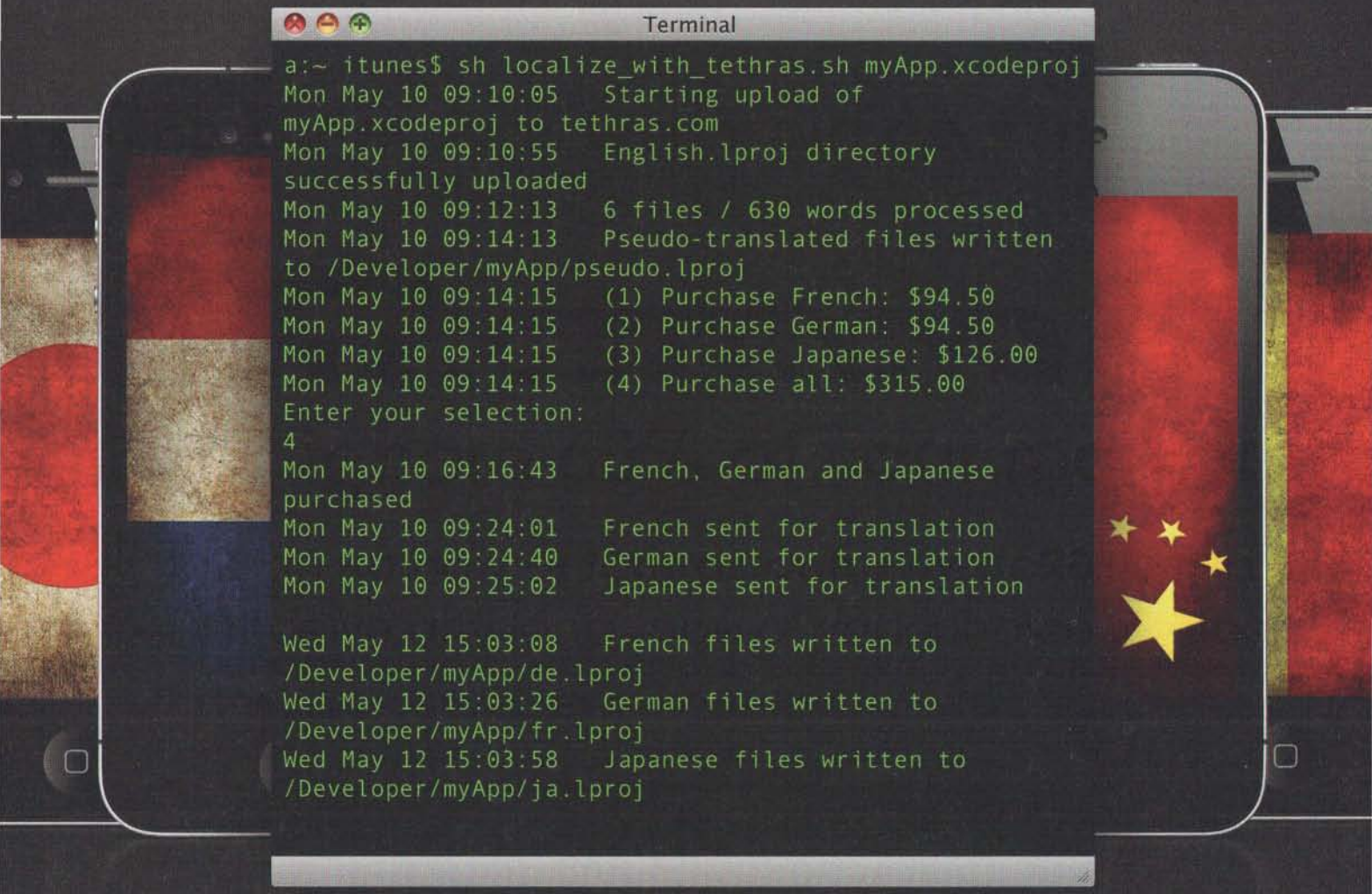
Figure 2 - Template selection.

Once you've chosen the template, Xcode has some more questions for you. Figure 3 shows this dialog, along with the values you should enter for this project. Feel free to enter a different value for "Company Identifier," of course.

DOES YOUR APP ONLY SPEAK ENGLISH?

Go Global and Localize with

TETHRAS



Don't ignore 60% of your market

Sign up now at **tethras.com/wwdc2011** and receive any language of your choice for free

Limited to new customers. Offer limited to $100 off your final purchase price for any language you choose. Register before end of July to activate your code.

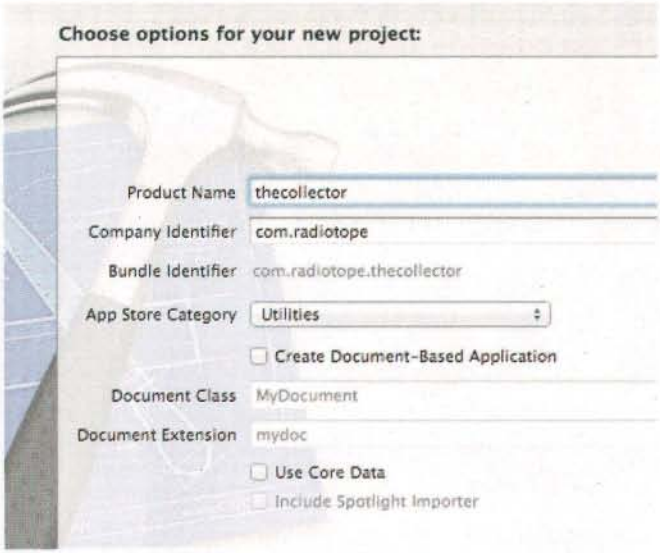


Figure 3 - Project options.

You can certainly leave 'App Store Category' at its default, but, this will be a utility, so, there's no harm in setting it.

Once these options are set and you click next, you'll be brought to the workspace in Xcode. Figure 4 shows the default workspace with your project options available.

In a column on the left, you'll see the navigator pane, defaulting to the project navigator. The project navigator shows all files (source code, resources, etc.) available to this project. With the project selected we're looking at project options in the main, center pane.

You may notice that just by creating the project, there's a lot done for you. In fact, you already have a complete, runnable application. Indeed, you can click the Run button in the toolbar right now and, after Xcode compiles the application, you'll see a blank window.

Now, you may not think that blank window is impressive on its own, but actually, there's a lot going on! First, even though it's blank, the window behaves as you'd expect: you can move it, resize it and close it. Also, note that you have a working menubar that also responds to the 'Quit' command. It is Cocoa that is taking care of these aspects for you. Over time, you'll learn to let Cocoa take care of as much as possible for you. Quit the application for now, using the menu or command-Q, and let's get back to Xcode.

Main Window

Of all the things that Xcode, and the MacRuby template, get set up for us are some basic files. Looking in the project navigator on the left, you'll notice 'AppDelegate.rb', which is where we'll put our code, and under the Resources folder, MainMenu.xib, which represents our user interface (the Menubar and window along with all UI components).

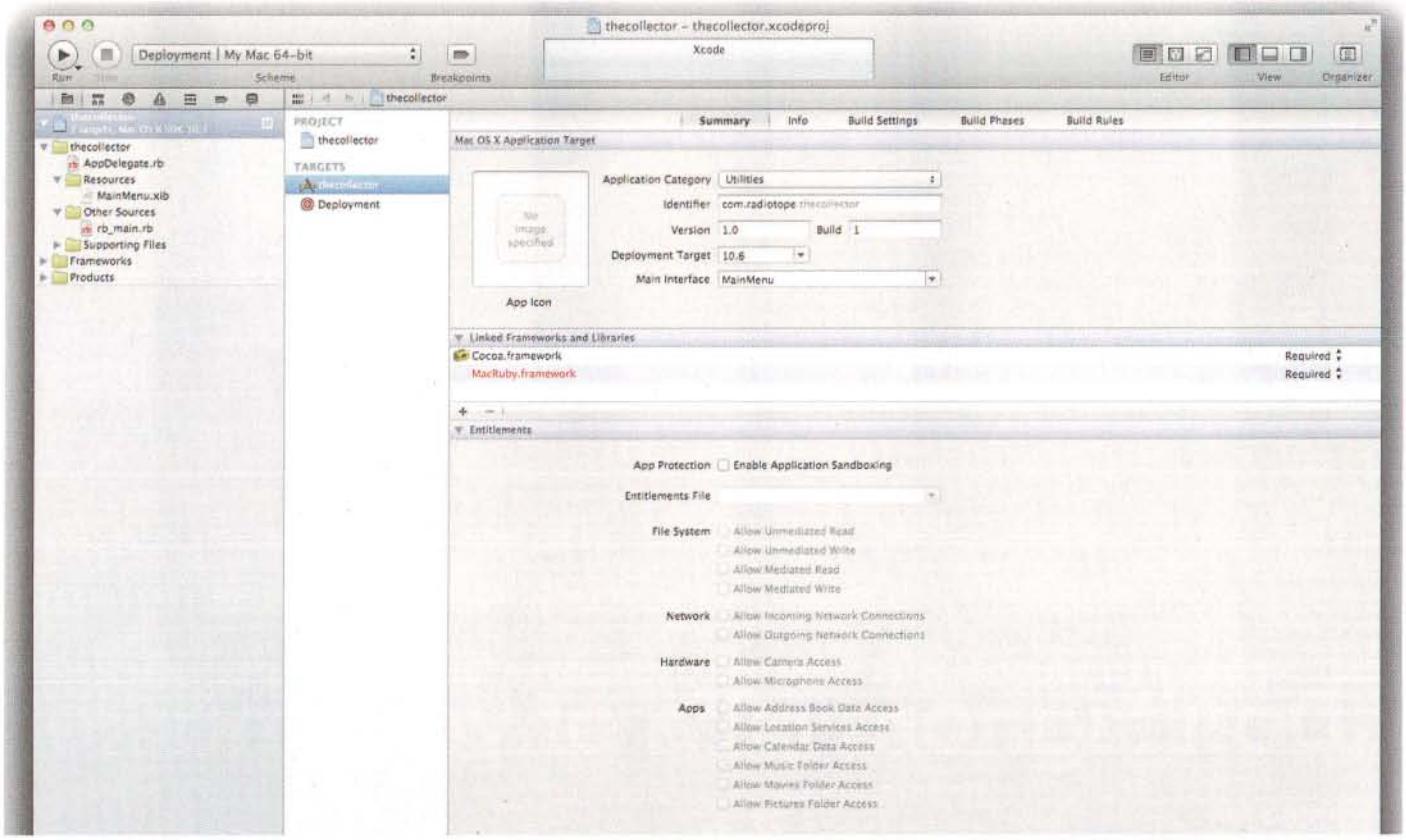


Figure 4 - Default project options

codefortytwo software

Back up everyone, everywhere.

CRASHPLAN™
Continuous Backup for Everyone

The easy, automatic way to protect all your data.

Individuals, businesses and enterprises around the world count on CrashPlan+, CrashPlan PRO and CrashPlan PRO*e* to keep their data safe.

You can too! **Try it for free for 30 days.**

Peace of mind is just a few clicks away at crashplan.com

Code 42 Software • one main st. se #400, minneapolis, mn 55414 • 612.333.4242

Framed!

One thing I've noticed with some MacRuby projects: the template doesn't properly pick up the MacRuby framework. We'll need this to run the application. While you're already in the right place—looking at the frameworks that will get linked in—highlight the red "MacRuby.framework" line under "Linked Frameworks and Libraries" and click the minus sign beneath it. Once removed, click the plus sign to add a framework. In the resulting dialog, shown in Figure 4a, type MacRuby to narrow the list, highlight the MacRuby framework, and click the 'Add' button.

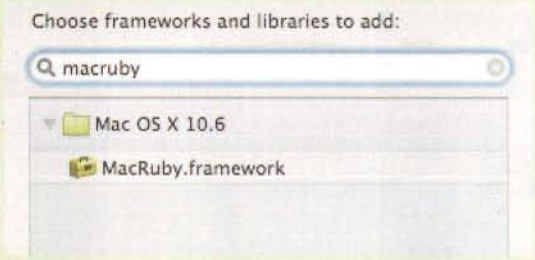


Figure 4a - Re-adding the MacRuby Framework.

This is also how you can add a framework when needed in a different project. Once added, this new framework will show up in the project navigator in the left pane. If you're tidy like me, you'll drag it from the default location into the 'Frameworks' folder.

Let's create our interface right now. Click (once) on the Mainmenu.xib file in the project navigator, and you'll be brought into Interface Builder, Apple's GUI for creating GUIs.

You'll be looking at a graph-paper inspired workspace with your application's menubar at top. If you like, go ahead and double-click on application name in the menu-bar and change it from 'thecollector' to 'The Collector'. Now, let's create a user interface. Click (once) on the object representing the window in the object column (see Figure 5).

Figure 5 - The window object

This will reveal the empty window that we saw earlier when running the program. You should now be looking at something like what is shown in Figure 6.

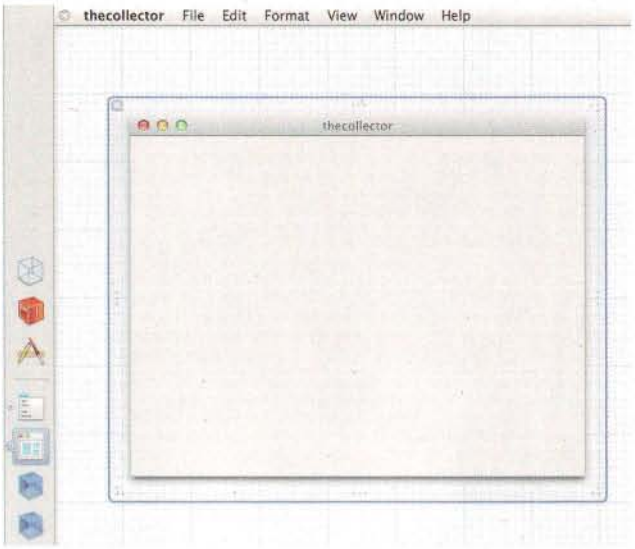


Figure 6 - Interface Builder displaying our window.

Now we need to place user interface objects on the window. You need to expose the palette of objects by clicking on the appropriate view button in the toolbar. Click the third 'right panel' button in the toolbar, shown in Figure 7.

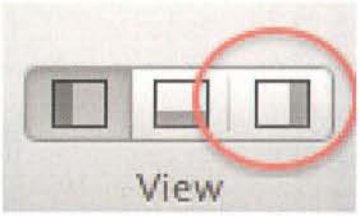


Figure 7 - Expose the utilities and object palette.

At the bottom of this panel, by default, you'll see the file template library. We want to select objects from the object library. Click on the object library icon (the cube), as seen in Figure 8.

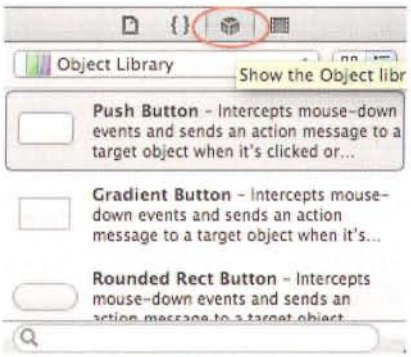


Figure 8 - Selecting the object library.

usedcisco.com
World's Largest Network Hardware Outlet

World's largest used Cisco outlet

Over 2000 of the most popular parts in stock

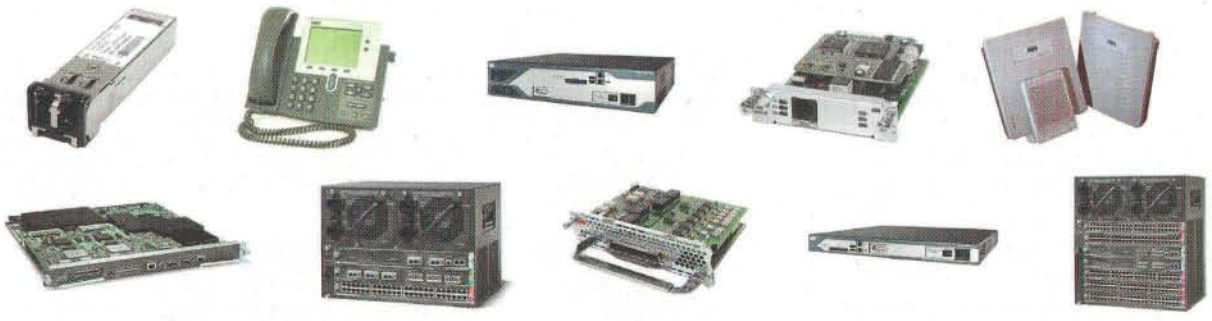
SAVE 60 - 90% off list

- **FREE 2 YEAR WARRANTY** with coupon code: MACTECH*
- **FREE SHIPPING** on orders over $200 (for a limited time only)

Call us: **800.504.7199**

Visit us on the web at: **usedcisco.com**

Just some of the parts we carry....



*To use your coupon, either: 1) Call in your order (800) 504-7199, and mention coupon code MACTECH OR
2) Shop online at usedcisco.com. On the last step of checkout, enter coupon code MacTech in the "Salesperson" field.

Complete your order online (you will not see your discount applied here). We will apply the discount when we process the order, and only charge your card the adjusted total. You will be a new confirmation as well. Coupon not combinable with any other coupon, discount or special promotion. Coupon not transferable. Coupon applicable to web pricing only.

Now we can choose some interface objects and place them on our window. You can make the selection easier if you use the search box at the bottom of the pane. (Again, see Figure 8.) Let's start with a label. Type 'label' into the search box, and you'll find there are two types of labels: a multi-line label and a plan, 'label.' Choose the multi-line variety: click and drag it from the object library window to the window the represents our application's window.

You should notice that when you drag an object to the window, Interface Builder helps you align it so it's positioned properly. It will have the right spacing from the borders of windows and away from other objects. Interface Builder does this with the blue guidelines it temporarily places on the window. Figure 9 shows this in action.

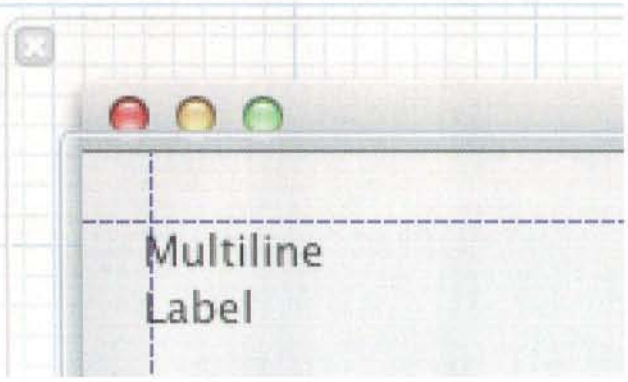


Figure 9 - Interface Builder displaying guidelines.

Place the multiline label in the upper left-hand corner of the window, using the guides for placement. Once placed, hover over the right side of the label until the cursor turns into the double-arrow cursor for resizing, and expand the label all the way to the right, until the guides show you the boundary. Next, type 'button' into the object library search box. Choose a push button and drag it to the window. Position it flush right and just under the existing label. Again, the guides will help you get the positioning just right. Figure 10 shows the completed interface.

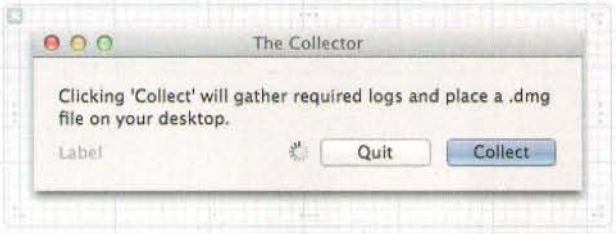
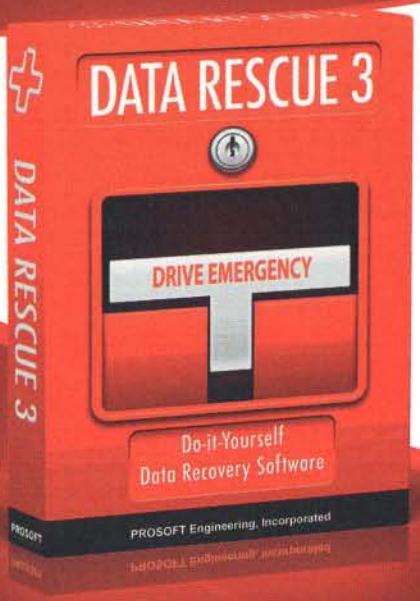


Figure 10 - The completed UI as shown in Interface builder

You've already dragged a multiline label and button to the window. This UI consists of the following additional objects—use any word from the object name to search for it in the object library (I've put the word in **bold** type that you can use in the search box):

A second push **button**

A circular progress **indicator**

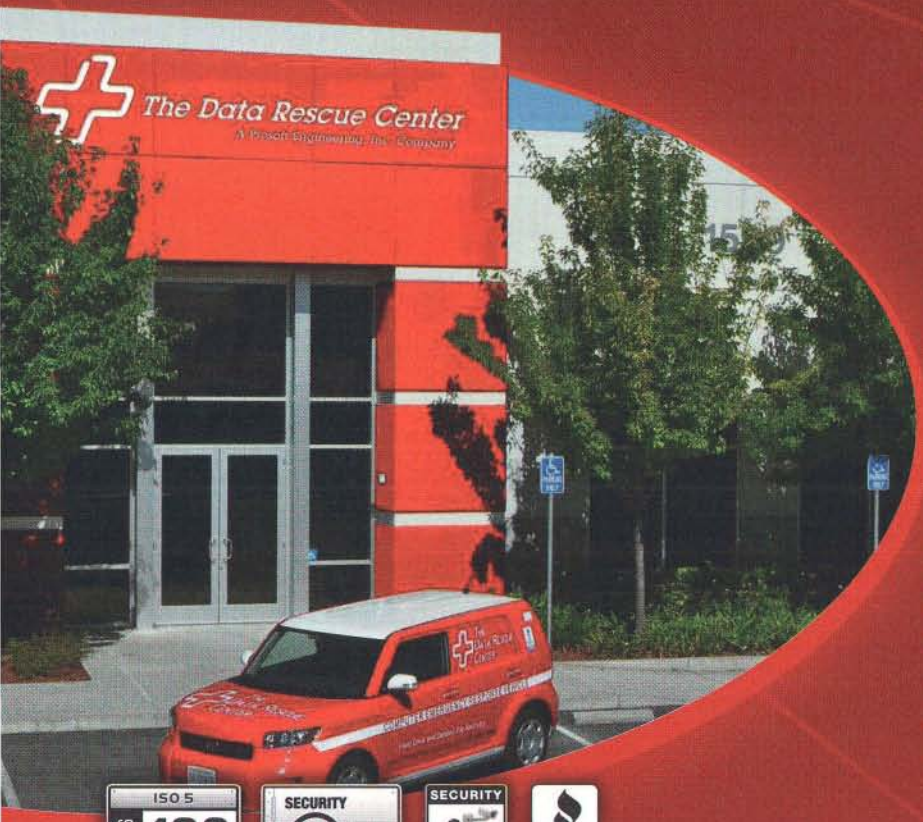


From the makers of...

Data Rescue 3*

The best selling data recovery software for Mac...

...comes the complete data recovery service.



The Data Rescue Center was founded by Prosoft Engineering, makers of the award-winning Data Rescue software. Our recovery know-how provides unparalleled advantages over other recovery services.

- FREE hardware diagnosis
- FREE specialized drive & laptop boxes sent direct to you at NO COST
- No Data, No Charge
- Professional Results at a Lower Cost
- Numerous awards in 2010 for Best Computer Recovery Service
- Class 100/ISO 5 Cleanroom
- Class 2 Vault
- FREE E-Waste Recycling

The Data Rescue Center is headquartered in Livermore, California at a new state-of-the-art facility near Lawrence Livermore National Laboratory.

This new facility provides our recovery engineers with the top technology to recover your computer data while offering best-in-class security features to protect your data from being compromised.

ISO 5 CLASS 100 CLEANROOM | SECURITY UL LISTED | SECURITY | BBB ACCREDITED BUSINESS

diablo BEST OF THE EAST BAY 2010 | The Best of Silicon Valley 2010 metroactive Best Hard Disk Recovery | BEST IN NORTHERN CALIFORNIA Best of the Bay 2010

The Data Rescue Center

hard drive recovery • data migration • photo archiving

1599 Greenville Rd | Livermore CA 94550

877-501-4949

TheDataRescueCenter.com

A **label** (plain 'ol single line label - stretch it across to the progress indicator).

Once the objects are in place, click and drag the bottom window handle to resize the window itself. Again, you'll get guides as you hit the right positioning. The objects themselves still have their default characteristics, so we'll also need to make a few tweaks to these objects. The simple ones first:

Double-click on each button to title it correctly (Figure 11 shows this in action).

Figure 11 - Giving a button its correct title.

Double-click the Multiline label and type in some appropriate text.

Now, choose the attributes inspector in the utilities pane. Figure 12 shows you this.

Figure 12 - The attributes inspector button.

Once the attributes panel is up:

Click on the window itself and change the title to "The Collector".

Click on the "Collect" button, and in the attributes panel, scroll down until you find the "Key Equivalent" attribute. Click on the Key Equivalent text field and press return. You should see a return symbol (⌧) in the text field.

Choose the single-line label and, in the attributes panel, look down the list and select (put a check in the checkbox) under View->Drawing->Hidden. We want this to start off hidden, and we'll unhide it in code.

Select the progress indicator and ensure that Behavior->Display When Stopped is *unchecked*.

That's it in Interface Builder for now. Let's add a little (actual) code.

Glue to Interface Builder

As part of Xcode, Interface builder has knowledge about the code we write. This allows us to interact with the code graphically.

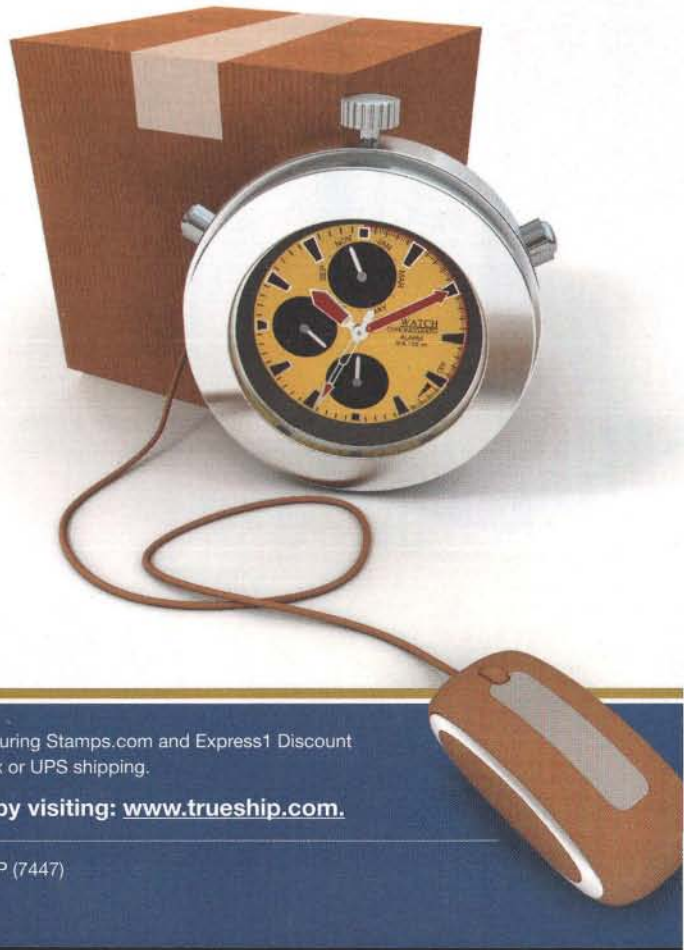
Ship Smarter.
Ship Faster.

ReadyShipper is the premier multi-carrier shipping software for the Mac.

ReadyShipper imports orders, creates shipping labels and packing lists and works with today's popular ecommerce systems. Get started today with a full-featured free trial that will have you shipping in minutes.

Free ecommerce plug-ins supporting these companies and many more.

Magento Open Source eCommerce
amazon.com
PayPal
ebay
network solutions
shopsite



Available only from TrueShip™

ReadyShipper is USPS ready featuring Stamps.com and Express1 Discount Postage with your choice of FedEx or UPS shipping.

Download your free trial by visiting: www.trueship.com.

Questions? Call Us: 877-818-SHIP (7447)

The Ultimate DVR for HD cable and satellite TV

eyetv

watch record edit enjoy

Award-winning EyeTV turns your Mac into the coolest television in the house. Work or surf the Web and watch TV at the same time. Record your shows to enjoy later, and play them on your iPhone® or iPad®.

eyetv HD

High Definition Video Recorder for Component Video

SRP $199.95

Elgato products are available at these and many other authorized Apple dealers:

Apple Store amazon.com MacMall OWC

elgato

Click on the 'AppDelegate.rb' representation in the project navigator. (Figure 13 shows this file in the navigator.)

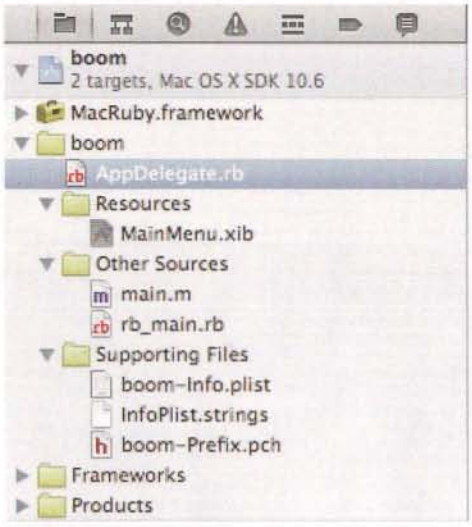


Figure 13 - AppDelegate.rb in the project navigator.

To begin with, AppDelegate.rb starts off with a basic shell:

```
class AppDelegate
  attr_accessor :window
  def applicationDidFinishLaunching(a_notification)
    # Insert code here to initialize your application
  end
end
```

This represents one class, named 'AppDelegate' that contains one method:

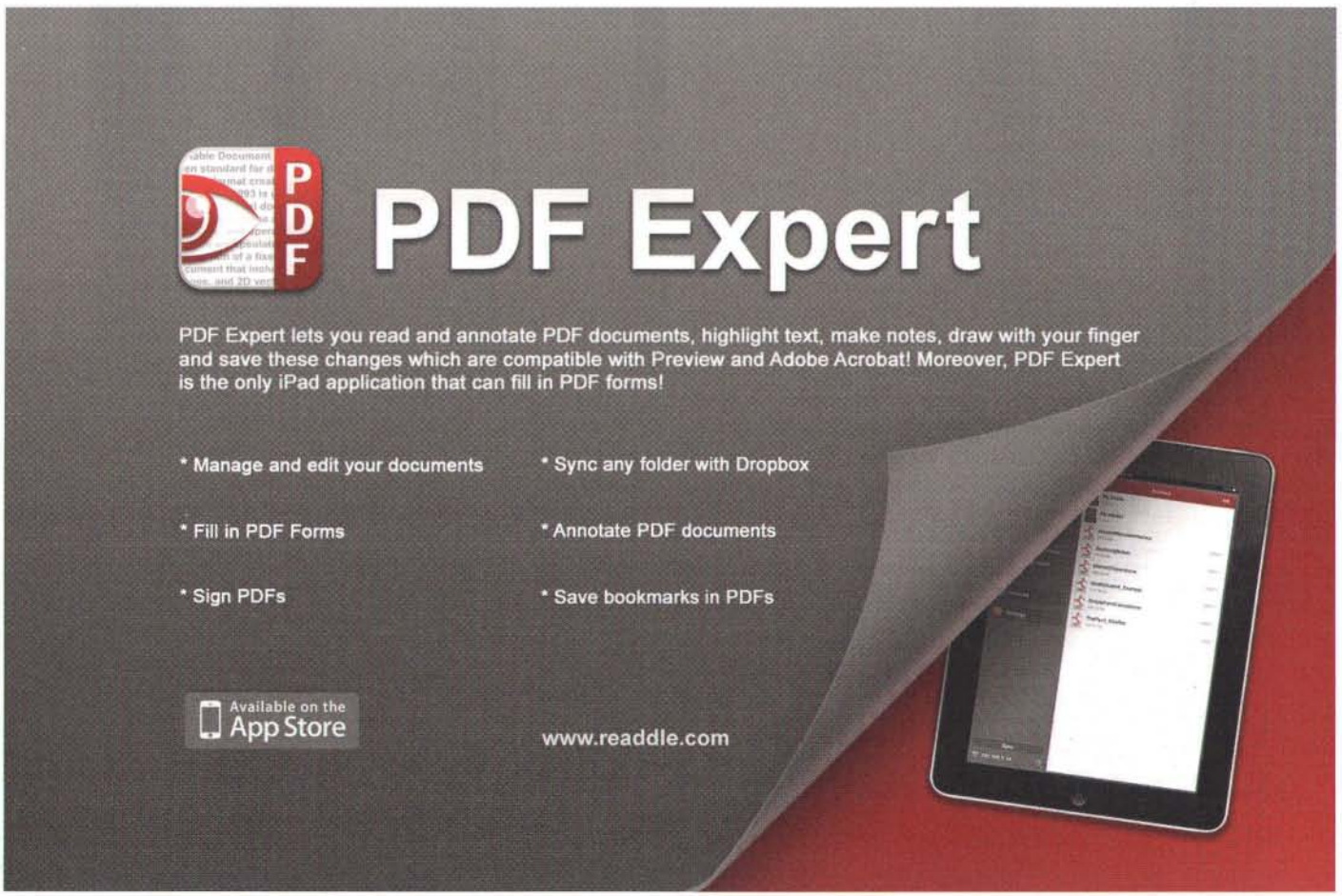
applicationDidFinishLaunching:(NSNotification*)aNotification. The applicationDidFinishLaunching method is one of those bits of magic that the Cocoa framework gives us. This is called by NSApplication once the application finishes launching. We're not going to use this particular functionality in this application, but do know it's there for you. Now, onto code that we will use.

You may notice that the first line in the AppDelegate class is a variable of type attr_accessor. Ruby provides a one-line way of creating an attribute accessor. You get both a *getter* and a *setter* for the given variable. (For more on accessors, see http://www.ruby-doc.org/docs/UsersGuide/rg/accessors.html.) These class attributes are also how Interface Builder knows how to tie into your code. They provide the glue, so to speak. The :window accessor provided by the template allows our class to interact with the predefined window. Let's create some for the rest of our UI elements.

Add the following lines to the AppDelegate class in the AppDelegate.rb file, directly following the 'attr_accessor :window' line:

```
# xib components
attr_accessor :quit, :collect
attr_accessor :status_label, :spinner
```

These attributes represent the two buttons, quit and collect, our single line label and our circular progress indicator. Next, let's add a class method so the collect button has something to take



audioengine

affordable audio!

Experience our award-winning sound, high-quality materials, and truly useful features.

BUY DIRECT ONLINE
free shipping, no tax
audioengineusa.com

$199 per pair

Audioengine 2 (A2)
Premium Powered Desktop Speakers

Closes the gap between computer speakers and home audio

*"These **are** the best speakers for your desktop, computer, or media player."* – Connect Reviews

Made for iPod

$99

Audioengine W2
Premium Wireless Adapter for iPod

$99

Audioengine W1
Premium Wireless Audio Adapter

High-quality wireless for your iPod, computer, subwoofer and all your audio gear

"Super-fast setup and the uncompressed sound is pretty remarkable." – Uncrate

Model A5N shown in Solid Carbonized Bamboo $449

starting at $349 per pair

Audioengine 5 (A5)
Premium Powered Bookshelf Speakers

Best-in-class active speakers for all your music

"There are no other speakers in this price range that come close." – Mac Observer

P4 passive speakers starting at $249/pair

$199

Audioengine N22
Premium Stereo Desktop Amplifier

Small yet powerful, with a high-quality headphone amp

"If you're looking for a desktop system that will give you great musical satisfaction with no hassle, you can't go wrong with the N22 and P4 passive speakers..." – Positive Feedback

Works with all your gear • Cables included • 30-day audition • 3-year warranty

Visit our website for more product info, reviews, and awards: **audioengineusa.com**

©2011 Audioengine Ltd. iPod touch not included. iPod is a trademark of Apple Inc., registered in the U.S. and other countries. "Made for iPod" means that an electronic accessory has been designed to connect specifically to iPod and has been certified by the developer to meet Apple performance standards. Apple is not responsible for the operation of this device or its compliance with safety and regulatory standards.

Presto! PageManager 9
The Ultimate Document Solution

- Stack files with various formats for easy page handling
- Wide-range of PDF conversion and features including searchable PDF
- The embedded scan module (TWAIN) supports a variety of scan option
- Document pallet for easy organization
- Document inbox for monitoring incoming files
- Filter the types of documents for quick view

Mac Universal

Presto! BizCard

Available on the iPhone App Store

* Supports iPhone 3GS, iPhone 4

- Capture business cards and convert to contacts
- NewSoft SFIT (Smart Field Identification Technology) distinguishes data fields of first name, last name, address, etc
- Simple and intuitive user interface
- Supports both portrait and landscape business cards
- Detects the orientation and automatically rotates the card image
- Captures multiple cards and performs batch recognition

NewSoft
http://www.newsoftinc.com

action on. (Effectively, what do we want to happen when the 'Collect' button is pressed?) Add the following code directly after the applicationDidFinishLaunching method:

```
def collectLogs(sender)
  puts "Running the collectLogs method"
end
```

While this doesn't really *do* much, let's go make it work.

Wire it Up

Go back into Interface Builder by selecting (clicking once) on the MainMenu.xib file in the project navigator. If you hover over the objects in the column along the left of interface builder, you'll notice that one of the objects is named "App Delegate". This is 'glued' to our AppDelegate class and its code.

We can send actions to or from objects created in Interface Builder to the code in our AppDelegate class. Like everything else in Interface Builder, this is done graphically. Interface Builder has gone through our code, looked at accessor variables and methods and lets us make connections. You need to point objects that need to know about each other, to one another. We do that with a control-drag (hold down the control key while you click and drag). You control drag *from* the object that needs to know *to* the object it needs to know about. Sometimes, this is a one way relationship—our status label, for instance, as we only set it—and sometimes it's a two-way relationship—for example, our Collect button, as it calls a method, and then that method can change the state of the button. Let's go do all of this now.

Attachment issues

How does Interface Builder know to attach this class object to our class named AppDelegate? You can see for yourself. In Interface Builder, click once on the App Delegate object (the blue cube named "App Delegate) and then select the Identity Inspector in the utilities pane. You'll see that the App Delegate is pointing to the class named "AppDelegate."

Fig 13a - Interface Builder pointing an object to a specific class.

We could point this to a different class (if we had one), or, we could assign other objects in Interface Builder to other classes. This isn't necessary in this simple example, but you can see how this allows more complex applications to be built.

SAINT® for Mac OS X

Integrated Vulnerability Scanning, Penetration Testing, and Checklist (Benchmark) Compliance.

Vulnerability Scanning

Assess any target with an IPv4, IPv6, or URL with pre-defined policies for PCI, HIPAA, FISMA, and more.

Identify CVE, OSVDB, IAVA, OVAL, and more.

Penetration Testing

Exploit vulnerabilities to gain remote access.

Run social engineering, phishing assessments, and more with the exploit tools suite.

Checklist Compliance

Show compliance with FDCC & USGCB security configuration policies defined by NIST SP 800-70.

For more information–

www.saintcorporation.com/mac

1-800-596-2006

SAINT is SCAP validated by NIST & is a PCI approved scanning vendor (ASV).

Copyright © 2011 SAINT Corporation. All Rights Reserved.

MacPaw

CleanMyMac

Fast and Clean. Easily and Safely.

CleanMyMac is the premier tool for keeping Macs clean and increasing system performance.

Get 20% discount!* Coupon Code: MTECHJ

macpaw.com/mactech

Macworld

For affiliate inquiries contact: affiliates@macpaw.com

*Limited time offer. Valid thru July 15th, 2011

Start by control dragging *from* the Collect button *to* the App Delegate object. Figure 14 shows this in action.

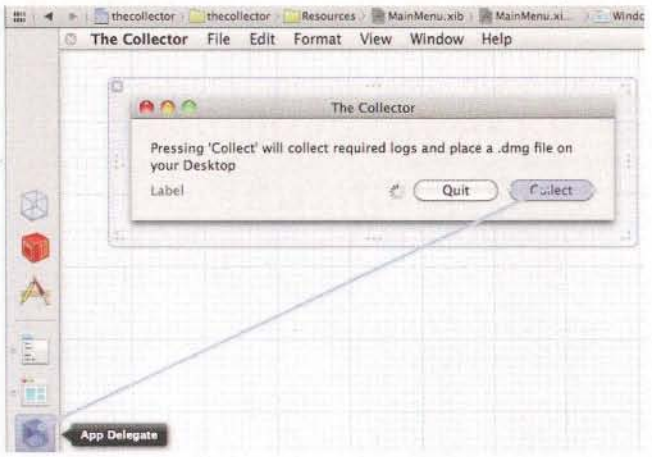


Figure 14 - Connecting objects.

Once you release the drag, the App Delegate object will display the actions it can receive. Since we've only written one method, that's all we're offered and can connect. Figure 15 shows how Interface Builder displays this. Click on the 'collectLogs' action in the pop-up menu to make the connection.

Figure 15 - Making the connection.

Now we need to also let the App Delegate class know about the Collect button. Control-drag *from* the App Delegate object icon *to* the Collect button. You'll find that Interface Builder doesn't know which of our *outlets* to assign to this object. An outlet is formed from our accessor variables, and allows our code to act on an Interface Builder object. When you release the drag over the collect button, you'll be presented with a pop-up menu that shows all of the outlets from out App Delegate class. Click on 'collect' to make the connection, as shown in Figure 16.

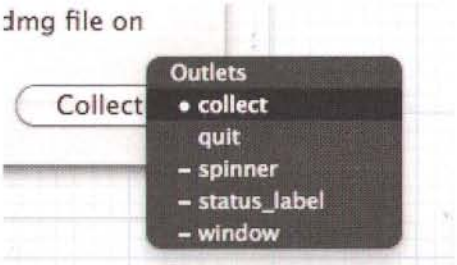


Figure 16 - Connecting App Delegate's collect outlet to the Collect button object.

Wish iCal were a Database?

Zulu iCal Server for FileMaker

Read / write to FileMaker live from iCal.
No synching, no scripting. Seamless.

www.seedcode.com/zulu

Special registration for current subscribers is: **http://www.mactech.com/bootcamp/subregister**

MACTECH® BOOTCAMP 2011

When and Where? MacTech Boot Camp, for those that support home users, small office, and small to medium sized businesses, is taking place around the country. The one-day, session-packed event will cover topics throughout the day. Lunch is included giving attendees the opportunity to not only learn from the best, but to also get to know others in the industry.

San Francisco, January 26	**SOLD OUT**
Dallas, April 27	**SOLD OUT**
Boston, May 18	**SOLD OUT**
Los Angeles, July 27	Registration Open
Chicago, August 31	Registration Open

One Track. Where MacTech Conference is for IT Pros and those supporting larger organizations, MacTech Boot Camp focuses on the needs of techs supporting consumer and small business markets. MacTech Boot Camp has a single track of sessions targeting what consultants and techs need to know to support home users, SOHO, and SMB. Sessions focus on both desktop and mobile, with appropriate levels of attention paid to the Mac, iPhone, iPad and iPod.

Full Array of Topics. Topics include: Marketing Oneself in a Community • Client Handling, Documentation, Passwords • Resources for Answers • Troubleshooting Hardware • Networking Basics and Troubleshooting • Printing Setup and Troubleshooting (Wifi, USB, Bluetooth, and Wired) • iOS Support • Windows on the Mac Options • Software Updates • Backup Systems and Options • Viruses and Security • Basic Scripting • Support Call Techniques • Remote Support and Access.

Lunch Included. Spend with peers you know and make new contacts.

Space is Limited. We have a limited number of event attendee spots available. Don't miss out.

Subscribers Get Special Pricing. While early bird pricing is available to start, current MacTech subscribers can take advantage of discounts during an extended window. ***But act fast! Even for subscribers, it ends soon.***

http://www.mactech.com/bootcamp/subregister

Now, the collect variable in code points to the button object we connected it to. Let's make the remainder of the connections. Connect the following *from* the App Delegate object *to* the object in Interface Builder:

status_label to the single line label.

spinner to the circular progress indicator.

We also get an action for free: quitting the application. We can quit in the same manner as the quit menu item. Control-drag *from* the Quit button to the Application object. This is an object that we haven't looked at yet. Instead of a blue cube icon, it has a 'generic application' icon (a pencil, paintbrush and ruler that form the letter 'A'). This pop-up menu will offer many more received actions. Connect the Quit button object to the 'terminate' action.

That's it! Run the application by clicking on the Run button in the toolbar.

First Run

You should now see the application running with the full user interface that we defined in Interface Builder. Further, it works! Well, it works in as much as we defined. The Quit button should work. The Collect button doesn't collect any logs, but it does do what we defined in the method it's pointing to:

```
def collectLogs(sender)
  puts "Running the collectLogs method"
end
```

All this method does currently is to write a message to the console that says, "Running the collectLogs method." You should see this message each time you press the Collect button. Where? In the console! You can see that by enabling the "Debug area." You can do this on a case-by-case basis by clicking on the 'Enable debug area' icon in the view menu of the toolbar (the middle of the three buttons). There's a way that I like more: enabling it on run and hiding it after a run completes. You can set this by opening Xcode's preferences and selecting "Behaviors." Figure 17 shows the setting to enable the debug area on run.

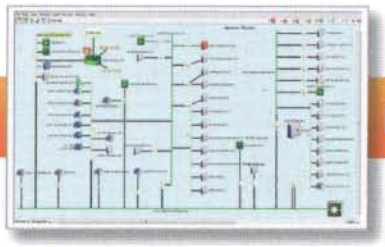
Figure 17 - Showing the debug area on run.

InterMapper

Real-time Network Knowledge

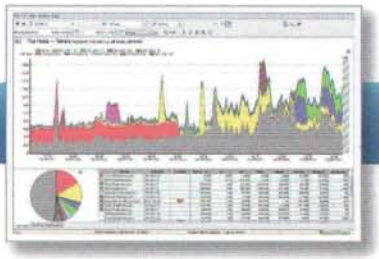
The #1 Network Monitoring, Mapping and Alerting Software for the Mac

Diagnostics



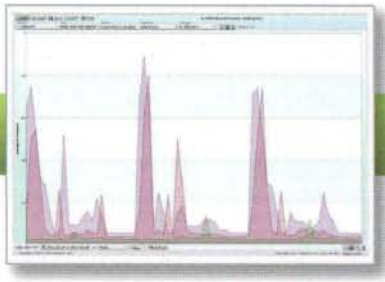
Layer 2 Connectivity

Troubleshooting



Flows Analysis

Capacity Planning



Reporting

- **Powerful**—Monitor anything connected to your network
- **Flexible**—Runs on Windows, Mac, Linux
- **Easy to use**—Simple configuration with auto-discovery
- **Affordable**—Fully integrated, no add-ons to buy

Download a Free 30-day Trial

www.intermapper.com

BUSINESS FOR SALE
Sedona Arizona

MACCLINIC
PCCLINIC

Authorized Reseller
Authorized Service Provider

- Established business, growing and expanding since 10 years
- The only Apple Authorized Reseller and Service provider in the area
- Prime retail storefront located on the main shopping street
- Wealthy tourist town, 3 million tourists per year, high-end customers
- Continuously achieving highest Apple Service Excellence
- Highly trained and extremely friendly staff

www.appleresellerstore4sale.com
info@appleresellerstore4sale.com

SOUTHERN STARS

SkySafari, meet SkyWire!

Your celestial travel guide. SkySafari puts 2.5 million stars and 30,000 deep sky objects at your fingertips — it's mobile astronomy for everyone!

Plug-and-play telescope control. SkyWire adds a serial port to your iPhone, iPad, or iPod Touch – and turns it into a 21st-century telescope controller!

Available on the App Store

Made for iPod iPhone iPad

www.southernstars.com +1.415.671.6251

You can automatically hide debug area by choosing the same option under the "Run completes" action, expect you'll set the pop-up menu to "Hide" instead of "Show."

Collecting Logs?

That's a lot to cover in one month. I'm going to hold the discussion of the full source code until next month. If you're feeling adventurous, add some code to the collectLogs method and see what happens. In the meantime, I'm going to put the working application and source code up on the MacTech ftp site at ftp://ftp.mactech.com.

Conclusion

We covered a *lot* of ground this month. You may have installed and run Xcode for the first time. You likely installed the latest version of MacRuby, as it's not only evolving fast, but needed an update to work with Xcode 4. Finally, you got familiar with Interface Builder and how it interacts with the code you write by hand. As I mentioned, the full, working source will be available right now on ftp://ftp.mactech.com. Look at it and poke around. The walkthrough will go through what, how and why of what we put in there.

Media of the month: Portal 2, by Valve Software (available on Steam). If you haven't played either Portal, do yourself the favor and get Portal and block out about 8-10 hours for some relaxation time. If you haven't played Portal 2, it's worth your time.

If you haven't seen, registration for MacTech Conference—taking place in Los Angeles on November 2-4 2011—is now underway. If you're a technology professional, you owe yourself a little professional development time and interaction with your peers.

Find out more at
http://www.mactech.com/conference/about.

Until next month, relax, soak in/up all of the WWDC news, get some more Ruby practice in on your own and don't be afraid to experiment!

About The Author

Edward Marczak is the Executive Editor of MacTech Magazine and writes the monthly "Mac in the Shell" column. He co-founded the MacTech Conference and has authored several books about Macintosh technology. Most recently, this includes Enterprise Mac Managed Preferences, published by Apress.

```
if (you_have_a_website == true) {

    measure_roi = easy;
    contact_visitors = yes;
    real_time = of_course;
    try_visistat = free;
    setup = no_brainer;

}
else {

    no_clue = true;
    i_use_google = sorry;

}

//REAL-TIME WEBSITE TRACKING
goto = www.visistat.com;
```

by Ryan Wilcox

Pricing (Part 2)

More thoughts on pricing

Introduction

A small business thrives on cash. Cash is needed to pay the bills, and often it's a balancing act between the cash coming in and how much one needs to work to obtain this cash.

In *MacTech's* Feb 2011 Cowboy Consultant article, I mostly focused on the expenses you have as a full time cowboy consultant, and how to take that number and arrive at an hourly rate (given billings of 20-30 hours a week). Factor in insurances, retirement, travel, some equipment, and shrinkage and you have a (naive) hourly number.

This article holds additional considerations for your hourly rate number. For example, what can the local market bare? But then again, don't price yourself too high. Or too low! Account for slack in your business: there will be times when you're not billing 40 hours a week (or even 30).

Also, when you figure out your hourly (or project) rates you also need to think about your goals in the future: goals for your personal success financially, in addition to your successes as a person.

Other Thoughts that may affect price

Brief review from last article

To rehash a little bit from Part 1 of the pricing article: your hourly rate depends primarily on how much, per month, it costs you to live in the area you're living in and how much work you can schedule in a given month. For example, if you have a $1,200/month mortgage, you have to make sure you're bringing in at least that much (actually, much more, when you factor in food and utilities), if you're a full time consultant cowboy.

Another thing that could affect price, especially if you're coming from a traditional employee job, is the number of things your employer used to pick up. You, yourself, are now 100% paying for your health insurance, liability insurance, and travel. Likewise, you'll also be spending money just to keep your business afloat: the occasional lawyer fee, advertising, and maybe web design if that's not your strong suite.

Your current debt, budget, and time for projects

Your level of debt would also affect your price also. The average credit card debt for a family is $14,000+, and the average student loan debt is $23,000+. The payments on these loans, of course, need to factor into your monthly budget. This means you'll have to bring in more money a month than someone without those loans.

If you haven't (or haven't!) sat down and made out a household budget, that's probably the first place to start. Then divide by 4 to get how much money you need in a week, and divide by 5 for money per workday.

Likewise, you need to pay for your business development time too: time doing bids for projects that don't pan out, time doing marketing and research.

Likewise, vacations, and any other kind of day off, are always unpaid. This is certainly something to think about when you're planning out your budget and monetary requirements for the year.

Of course, all this needs to be balanced with the client project work on your desk currently. Sometimes even this is a balancing act: too much client work and important business development (which means project work in the future) goes unattended. Too much business development, and rent doesn't get paid. (More on this topic latter in this article)

Minimum Viable Income

"How many clients do you need to serve to make even a crappy living?" - The Intelligent Entrepreneur by Bill Murphy Jr.

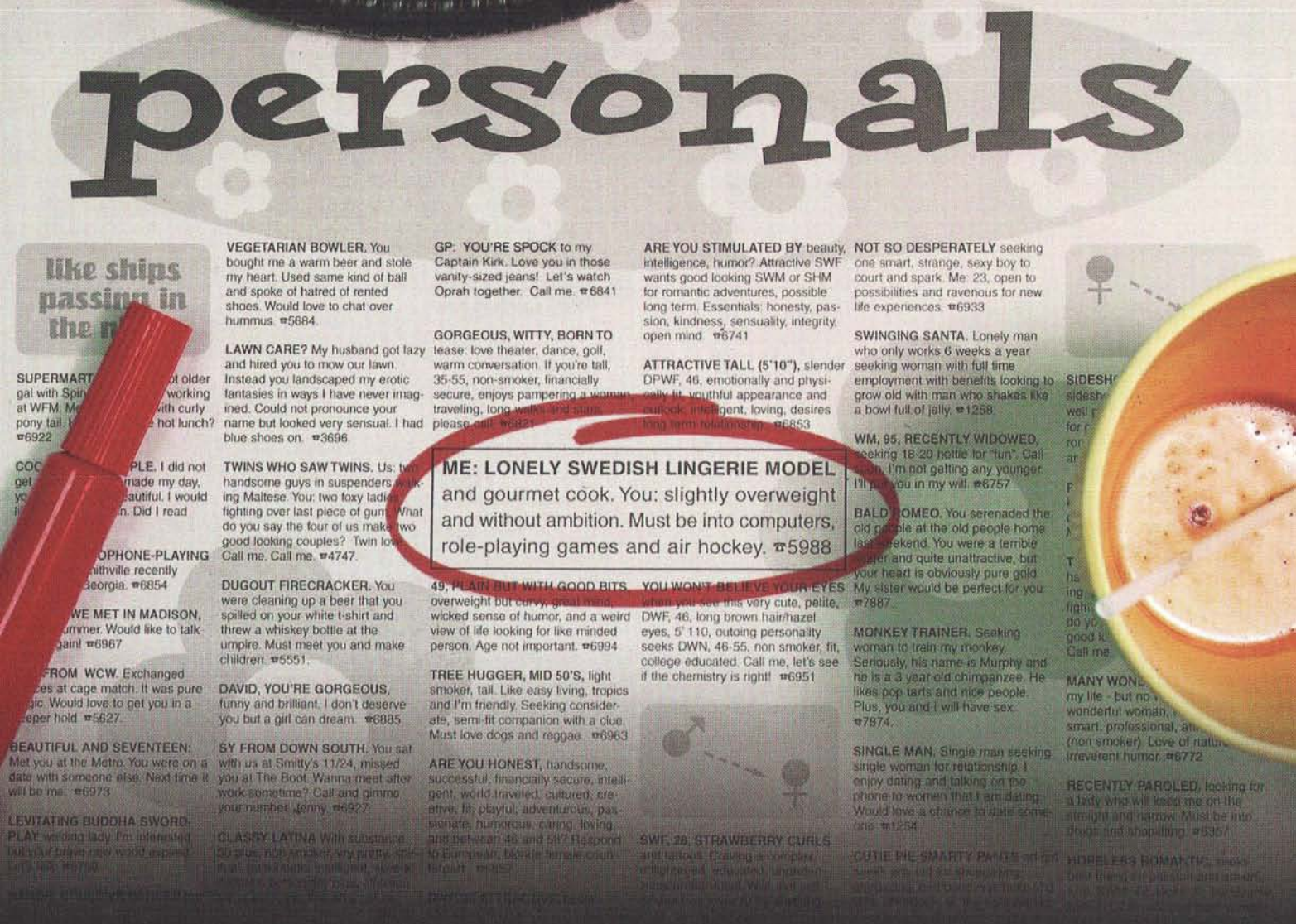
While, *The Intelligent Entrepreneur* focuses mainly on startups that got big, that bit of advice applies to freelancers also. How many clients (or projects, or billable hours) do *you* need to make a crappy living?

In an Agile software development project, often the question comes up: "What is the minimum viable product that we can ship and get out to people?" As a cowboy consultant a similar number should be in your brain: the smallest amount of money you really need to survive. Then strive to make that, and strive to make twice that, and so on.

What will the local market bear?

I've previously talked about the advantage of living in an area with a low cost of living. The advantage is the little amount of money it takes to live. The disadvantage? Local businesses can't charge as much for their products, and thus might not be able to afford to buy your services at a high hourly rate.

Then again, you also have to make enough money to live. In a big city area you have another problem: dealing with competition. Your clients may be expecting to pay within certain price range, which may (if you're lucky) be set by your competition. "I paid $500 for this service at Brand X Consulting



We're Easier.

In fact, Real Studio is the easiest, fastest way to create software for Mac OS X, Windows, Linux and the web.

Why use Real Studio? Real Studio is the only object-oriented, cross-platform software development tool that enables users at all levels to create powerful, stand-alone, native applications. With over 40 native user interface controls including buttons, lists, fields and tab panels, extended database support, native reporting and Internet and networking features, Real Studio is cross-platform that really works.

Now, Real Studio Web Edition allows you to use this powerful development environment to easily build web applications — no need to know HTML, JavaScript, CSS, AJAX and PHP. And, unlike those usual web technologies, Real Studio compiles your web applications to binary, so they are safe and easy to sell.

Real Studio. Cross-platform development for humans.

real.studio

For a limited time, getting Real Studio is easier too.
Get 20% off when you buy a new Real Studio license.
Use coupon code WWDC2011.
www.realsoftware.com/realstudio

last time I did it, so that's about what I'll pay this time" might be what your client thinks.

The articles on marketing will talk about differentiating yourself from the competition, but there's also a range: price too high and potential clients can't afford you.

Your 3-4 month stash

Ideally your prices will also let you build up a stash of money you can use when you are between projects, and replenish this stash over time. I like a 3 month stash: this gives enough time for 1 month of project hunting, 1 month of working, then another month of waiting to get paid (assuming NET-30 terms on the invoice).

In my experience even the simplest projects take one calendar month to finish, and another calendar month to get the money. The first month, from initial contact to completion of project, usually goes like this for me:

- Week 1: Initial contact, I draw up a quote, they consider it and say yes
- Week 2: Initial Implementation
- Week 3: Client QA, and making fixes on said project
- Week 4: Deployment and final tweaks.

While my experiences lately have been with these as small side projects, some of this time delay is unavoidable. You might be laughing at me now, but let's assume 1 month to be on the safe side.

The kicker here is often NET-30 style terms on the invoice ("I give you 30 days to pay this before I get annoyed"). With this calculation it's 2 months from the initial concept to when you get paid: one or two months rent. What money do you use to pay those bills while you're waiting for that money to come in?

This is the good scenario too: when the project concludes successfully and both parties go their separate ways, happily.

However, note that I've had to pay rent twice since we said, "yes" to the contract? This delay could mean some tight financial times as you wait for that check to come in, as bills pile up. This is where the stash comes in, to give you a little boost through those lean times. Your hourly price should be high enough so you can pay back that stash occasionally.

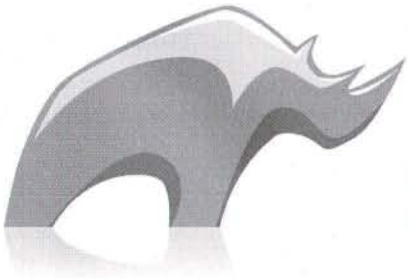
Don't price yourself too high

As a computer programmer I realize that programmer time is expensive: it's a highly skilled profession, and big complex applications take time to build, thus expensive. I do what I can to keep my services affordable to potential clients.

Lower cost means more people can afford my services, and I can build more long-term clients... or so my theory goes. I'll grant you that this may or may not be correct, but that's my theory. I also, personally, want to make sure small businesses get the technology solutions they need. That's a market I'm interested in (I'm a small business, I feel I should help other small businesses), so it's important to me to be priced appropriately.

Don't price yourself too low

To counterpoint everything said about pricing too high, you also don't want to price yourself too low. There are two reasons for this: budgetary, and appearance.



LassoSoft inc.

security. speed. simplicity.

Proof that all programming languages are not created equal.

Quick to learn, fast to deploy, Lasso is a powerful, 'batteries included' development platform specifically geared to the web that has 95% of the libraries you'll need to do the job, and db connectors, email support, curl support, SOAP, PDF, and endless scalability from C and Java API's - all included right out of the box.

Free full product download for learning and dev builds. Full license from $49/month leased or $399 to own. Extensive community and free resources.

www.FADIGEAR.COM

Audioglove

Protective iPhone Cover

Finally, a case that rocks just as hard as you do!

How it Works?

The **FADI** protective carrying case, also known the Audioglove, is designed in such a way that the user can open the case and redirect the sound toward him/herself. This audio enhancing functionality is not offered on any other protective carrying case. The Audio Glove enhances and amplifies the volume and quality of sound emitted by the iPhone.

It's simple design corrects a fundamental flaw in the iPhone that suppresses good quality sound and tone production. The retractable audio drawer on the FADI Audioglove eliminates the need to attach external speakers. Now you can amplify the phone without restricting its mobility and ease of use.

As an added bonus, the Audioglove makes the speaker-phone sound warm and full for people on both ends of the call — now your iPhone can operate as a hands free device without a Bluetooth accessory!

Constructed from polycarbonate - The same material used for bullet proof dividers at banks.

www.fadigear.com

Audioglove available for 3G/3GS iPhone 4 (Verizon and AT&T)

Budgetary concerns are the kicker. As a consultant you should not assume you're going to bill 40 hours a week. There's a balance here: you have certain expenses every month (rent/mortgage, taxes, utilities, food, savings) and only certain number of hours in a week to get this all done.

The next concern is cost vs. quality and perceptions. A low hourly rate might actually drive away potential clients! Potential clients might scoff at your rate and think, "I need a professional to do this work, but at this price I'm obviously getting a kid right out of school, who wouldn't be able to handle this kind of task".

A low rate might actually come at the cost of your own quality. For example, a low rate might mean you have to bill 50 hours a week to make rent, leaving no time for personal development and enhancing your skillset. In technology this is important: going to conferences or reading, or open source, volunteer, or writing work.

So, while your rate might seem extraordinarily high to you, if you can fill those hours then it's just about right.

Other, less obvious signs, might point to your rate being too low:

- You live modestly, but there never seems to be enough money to make ends meet. If this happens several months in a row, time to reevaluate something. Maybe you're in the doubly-taxing position of having too much work, but struggling to pay rent (or taxes). I've been there myself - it's really no fun. Time to step back, evaluate what's going on.
- You feel like you need to work more than 60 hours a week, most of the weeks of the year, to pay the bills. While you probably won't get rich as a Cowboy Consultant, it also shouldn't land you in the poorhouse.
- You seem never to have quite enough to pay your taxes, or you have to set aside an entire month's worth of income, at once, to cover your quarterly taxes.

Hours you can find to bill in a week

I talked about this in the "Price for =< 20 – 32 hours a week" section of the first article. Each of us has responsibilities outside of billable work. For example, business development is an important part of running your business: you need to spend time to bring more clients in. This takes time.

Likewise, you might have other commitments outside billable client work. I assume that I'll lose one day a week to errands, writing, and my own projects, so I adjust my rates accordingly. Taking one day off a week gives me time to do other, unbillable things. I'll use Wednesdays to do estimates, run errands, work on some of my side projects, work on business development or personal development. It also helps me, personally, keep my sanity.

The Wealthy Freelancer (by Slaunwhite, Savage, and Gandia) talks about time management like a puzzle: you can swap out a piece here and there and replace it with another piece, if you need to. For example, occasionally I'll take Wednesdays and spend half the day on client work, to finish up

a big project. That means something (business development, for example) loses out. Or maybe cutting back on something work related to be able to pick your kids up from school.

Some weeks you'll just get less work done than normal. That's where the 3-4 month stash is also useful to cover some of this slack. That's also the tricky thing about billing based on hourly rates: sometimes you don't get in as many hours as your budget requires. (The next Cowboy Consultant article will talk about different billing methods, which may have less of an impact on your budget!)

For example, I spent a fair bit of time over the past to weeks billing only about 20 hours a week, because I was working on business development. The business development needed to get done: I've been ignoring some of those things for 6 months or more, and I'm glad I have a little bit of a stash to cover for the hours I didn't bill.

Note: your income is limited by a scaling problem

Eventually you'll hit the issue where you realize there's only one of you, and that you can only serve a couple of ongoing projects at a time. If you're already maxed out on workload, you can't take on another client: there's a limit on your time.

This limit on your time means so you can't lowball 2 clients then think "I'll just take on a third project to pay my rent — that will make up the difference!"

Lowballing the price on something then hoping to make it up on volume doesn't work with real world goods, and really can't work for consultant cowboys. You can hire more people, but that has problems all its own.

From 37 Signal's *Rework*: "maybe the right size for your company is five people. Maybe it's forty. Maybe it's two hundred. Maybe it's just you and a laptop. Don't make assumptions about how big ahead of time. Go slow and see what feels right: premature hiring is the death of many small businesses."

Sometimes you will have to turn down that big job, because you don't have the 5 people it would take to deliver it. Maybe you want to take the contract, but it's a 3-month full-time contract, and you don't have the hours available for that. There are ways around that (teaming up with another freelance company to split up and finish the work, for example), but sometimes you have to turn down work because you're already booked.

Slack

There's also intentional and unintentional slack in businesses. For example, if you pack your calendar tight with projects, one being scheduled right after the other is done, what happens when one project slips a due date?

Better to create some intentional slack: end a project on Friday, and set your schedule so that you start a new one 3 or 4 days later. How much slack depends, but you do have to make sure your hourly rate is high enough so you can, in fact schedule slack.

PACKED!

The MacTech DVD - Volumes 1.01-25.12
is packed with more than ever before -- over 3200 articles from more than 290 issues **(1984 - 2010)** written by over 900 experts, all 29 issues of Apple's develop, 21 issues of FrameWorks magazine, 100+ MB of source code, MacTech Viewer, working applications, full documentation, demos for techs, ***and more!***

Everything is displayed in the very fast, very searchable ***MacTech Viewer!*** An application that has been designed specifically with Techs in mind. Search quickly through 25 years of great information provided by MacTech. Information to save you time, and make your life easier.

Mac Universal

Call Toll Free 877-MACTECH,
Outside US/Canada: 805-494-9797
http://www.mactech.com/dvd

Requires Mac OS X v. 10.4.5 or later

Unintentional slack can also happen. This might be the time it takes you to find the next client or project.

But, during both of these times you aren't working!

Your unbillable percentage

You might find yourself putting in more unbillable hours then you'd like. Or maybe you've gone a few weeks and not able to get the billable hours in that you'd like.

Maybe now is time to take a step back, and track *everything* in your day. How long you spend on email when you get up, how long you work on client work, how long you work on other stuff at night. How long you do marketing or machine maintenance, or other business Spring-cleaning that you might have to do? Do this kind of tracking for 2 weeks, and examine your habits. Maybe you spend two hours reading news every day, or too much time improving your own website. A little bit of tracking, and a little bit of analysis, can bring your time sinks to light.

If you don't like manually entering everything into your time tracking tool, try an app that does it for you! Apps that come to mind like this are: TrackTime, and SLife, but there are at least a half-dozen other apps for this on the Mac App Store. These will, at the very least, report that you were in NetNewsWire for 4 hours yesterday, instead of being inside your text editor, Xcode or whatever other app you make money from being in all day.

Dreams: Where you want to be

Beyond the basics of retirement, do you have other goals for yourself or the business? For example, going back to college, paying off student loans, taking on employees? This should also go into your budget.

Likewise, I poured my own money into my business initially, and feel the need to pay that back.

Depending on how close you are to retirement age, and what your goals are during that time, you might have to consider this in your rates also.

However, the awesome thing about freelancing is that you can set your own hours. If you want to "retire" and bill 10 hours a week to bring in some "pocket money" (which, at good rates, might actually be considerably more than simply pocket money.

Lifestyle design / Criteria for success

An alternative is to consider what you want your lifestyle look like. Instead of looking at your income and deciding what you can do with the money you have created, look at what you want to do and figure out how to get there.

Some of these thoughts will affect your prices. If you know you have only 11 months out of the year to make money in, because you want to take one month off every year, that's a factor in what you have to charge your customers.

The Wealthy Freelancer has a worksheet that encourages you to envision your ideal day, your perfect workday. Perhaps your workday includes a one-hour lunch

BMS

THE LAW OFFICE OF
BRADLEY M. SNIDERMAN

Helping clients with their software legal issues.

- **Trademark and Copyright Registration**
- **Trade Secret Protection**
- **Licensing and Non Disclosure Agreements**
- **Assist with Software Audits**

I am an attorney practicing in Intellectual Property, Business Entity Formations, Corporate, Commercial and E-commerce Law.

Please give me a call or an e-mail. Reasonable fees.

23679 Calabasas Rd. #558 • Calabasas, CA 91302
PHONE 818-706-0631 FAX 818-332-1285 EMAIL brad@sniderman.com

PDFpen: Feature #2 – Redact Private Information

New! Version 5

The affordable PDF toolkit

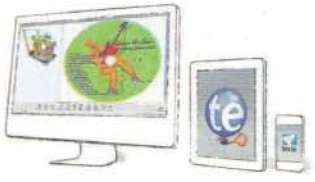
Can your PDFs keep a secret?

When you need to send out PDFs with private information, such as tax forms or medical records, you can control what gets shared—and what doesn't.

PDFpen lets you fully redact and remove text from your PDFs, not just black it out! Keep your PDFs buttoned up by making sure your private information stays private.

To see all the top 10 features of PDFpen, visit us at: www.smilesoftware.com/mactech

Smile



Software that's just right

Copyright © 2011 SmileOnMyMac, LLC. SmileOnMyMac, DiscLabel, PDFpen, PageSender and TextExpander are trademarks of SmileOnMyMac, LLC.

break allowing you to go to the gym everyday. Perhaps you need to stop work at 4 PM when your children get home from school.

Now think: does your lifestyle design change your availability, and given my income requirements how much do I have to charge my customers per hour?

For example if your lifestyle design includes scaling down your work during the summer to relax with your family, one way you can achieve that goal would be to set aside the mornings for work and then enjoy yourself in the afternoons. However, because your availability has decreased, the amount you charge per hour needs to be higher than if you were planning on 35 hours per week 52 weeks a year.

The *Four Hour Work Week*, by Tim Ferriss can help you here. It has excellent resources for lifestyle design and offers a "dreamline" worksheet, which gives you number crunching power for your ideas.

Now, don't expect to have all your goals, or achieve your ideal lifestyle, all in the first year, or two...or five. Setting goals now does give you some idea what your prices should be, and may let you understand yourself so you can pick the correct projects for you.

In the first article in this series, I talked about making a Criteria For Success list. Does any item on that list affect either the lifestyle you want, or the availability you have?

If you don't have a Criteria For Success list, now is a good time to make one. Ask yourself: "What do I define as a success?" This both helps you to define success, and gives you goals to shoot for. It's rather silly to have a goal, "I want to spend 3 weeks backpacking every year", but have your pricing or availability exclude the chance of that happening!

Maybe you don't have any particular goals. There's an interesting blog entry from the Dumb Little Man blog: (http://www.dumblittleman.com/2009/09/how-to-set-goals-when-you-have-no-idea.html) on this very topic!

Don't be Daddy-Who-Is-Boring

A Business Insider post caught my eye the other day: (http://www.businessinsider.com/the-wake-up-call-2010-12). This post talks of a man who started his business to be able to see his family more, who now his daughter calls "Daddy-who-is-boring".

The Work/Home dividing line pretty much disappears when you are working from home, out of a spare bedroom. Because you'll snatch up the laptop and do work while your kids are playing, or watching a movie, or hanging out - and there are you missing the action because a PRIORITY ONE email message came in.

Maybe you want clients that don't have 9 PM meetings Sunday nights (true story). Or have the ability to go to the park some days, instead of working.

This might involve saying no to a few projects (super late projects where everyone's going 80 hours/week, anyone?), or making sure to balance your time well... or maybe just leaving the laptop in the office sometimes.

Conclusion

You could see figuring out your hourly rate as way to value with both parties in the equation: value your client (by making sure they get you at your best), and valuing yourself (making sure you live a comfortable life - maybe not luxury, but ideally comfortable...maybe not at first, but eventually.

There is a careful balancing act here, one that needs to be reevaluated every so often (once a year, as part of your business retrospective, for example).

Next article will cover some other traditional and non-traditional methods of billing and payment, potentially changing some of your current client payment options.

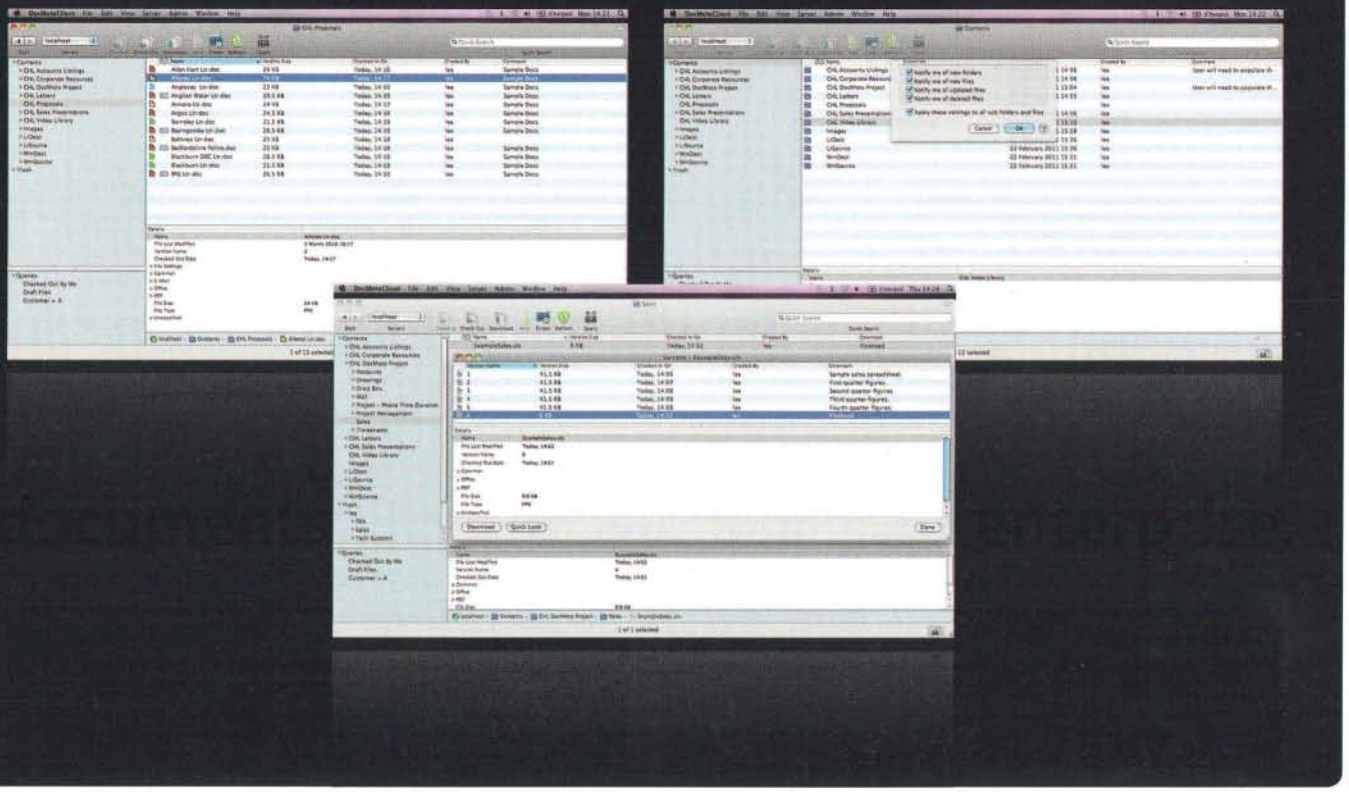
Create Outstanding Apps That Interface To Everything Musical

www.iConnectMIDI.com

About The Author

Ryan Wilcox has been consulting on his own for the last 8 years, through ups and downs in his business. In 2009 he started thinking about best practices for business, in addition to his normal thinking about programming. He can be found at: http://www.wilcoxd.com. Have thoughts or want to give feedback on this article? rwilcox@wilcoxd.com

Manage, share, search and catalog documents and files across your enterprise.



DocMoto is a professional, high performance document and file manager for enterprises using Mac.

Key features:

- Delivers managed collaboration on files and documents
- Retains a full revision history on all assets
- Powerful document and file searching
- Supports user configurable file cataloguing
- Ideal for all business sectors
- Fully native Mac interface
- Cost effective cloud or on-prem options available

For more information please visit our website: **www.chlsoftware.com**

CHL Software
Email: info@chlsoftware.com
Tel: +44 1242 225230

docmoto MAC

Developing Enterprise-Friendly Software for Fun and Profit

by Greg Neagle, MacEnterprise.org

MacEnterprise.org
Mac OS X enterprise deployment project

Introduction

Interest in Mac development seems to be at an all-time high. Sales of Mac hardware are growing at a rate faster than the industry as a whole, and many developers are hoping to build on their experience and success on the iOS platform by also developing for the Mac platform.

The advent of the Mac App Store has also brought new developers to the Mac. The Mac App Store solves a lot of problems for small developers, allowing them to sell their software to Mac users without having to develop and maintain a web storefront on their own.

Small developers new to the Mac may think of the Mac as in use only by individual users or possibly very small businesses, and might not give much thought to the issues around purchasing and deploying their software in large organizations. Even large, established software developers have been known to ignore these issues, and have shipped software that was not "enterprise-friendly."

Let's define that phrase. Software can be considered enterprise-friendly when it can be easily purchased for large numbers of users, installed with automated tools, and when installed, it works as expected for all users of a given machine with no additional post-install configuration that cannot be done as a non-privileged user (i.e., a user without administrator rights). In this context, "enterprise" really refers to scale. Any organization with a large number of computers can be considered an "enterprise" environment. Schools and universities face many of the same issues as private enterprises when it comes to purchasing, deploying and supporting Mac software.

In this column, we'll look at some of the common issues that make software purchase and deployment difficult in enterprise environments and what software developers can do to make their software more enterprise friendly.

Why bother?

If you are a software developer, why should you bother making the purchase and deployment of your software friendly to enterprise environments?

The number one reason is that if you do, you'll sell more of your products! If you make it too hard to purchase and deploy your software in an enterprise, systems administrators will look elsewhere and recommend other software that they can deploy in their organization.

Another reason: if your installer is not enterprise-friendly, and if their organizations insist on using the software anyway, systems administrators will almost certainly have to repackage your software in order to deploy it to their organizations. This introduces opportunities for errors to creep into the install process, as a repackaged installation may not install everything as intended, or may install things in a way that introduces problems. As an example, repackaging Adobe CS2/CS3/CS4 products often resulted in installation packages that stomped on or overwrote activation and licensing for other Adobe products. It's much easier to support your own software when you know it was installed with your own installer, and not via some hacked-together method developed by an overworked system administrator who is not as familiar with your software as you are. Repackaging software for installation introduces new variables to support. Worse, each organization that repackages your software may do it slightly differently – your support burden potentially increases each time a new organization purchases your software.

Making software truly enterprise-friendly should not be something left for the end of development when you are building the installer. Design decisions made early on can affect how easy it is to install, configure and use your software in a large organization. Being aware of the issues should help you make better design decisions.

Purchasing

The road to enterprise-friendly software starts with purchasing. Make sure large organizations can buy licenses for multiple installs/users of your software. Enterprise

STILL SOLID. WAY COOLER.

Keep your computers indestructible, free up crucial IT resources, and reduce support costs by 63%.

Faronics Deep Freeze automatically restores workstation configurations with every reboot.

For more information, visit
www.faronics.com

Faronics™

Available for:

administrators don't want to have to purchase 100 boxes of your software to install 100 copies, and they don't want to have to enter 100 registration/license codes and keep track of which machines have which license codes. Plan for some sort of multi-user licensing if you want to sell your software to large organizations.

Mac App Store

With the release of Mac OS X 10.6.6 in January 2011, Apple opened the doors to the Mac App Store, an attempt to bring the popular success of the App Store for iOS devices to the Mac.

The appeal and ease-of-use of the Mac App Store has enticed some developers to make their software available solely through this channel. The Mac App Store is a fantastic tool for individual users to discover and purchase software. But it is not very useful when it comes to purchasing software for enterprise use, as the terms of use for software purchased through the Mac App Store would prohibit the installation of a single purchased copy on hundreds or thousands of an organization's Macs. Add to this the fact that applications purchased through the Mac App Store are linked to a specific Apple ID – again, not an ideal situation for large-scale deployment. If you wish to sell your software to large organizations, you must provide a purchasing and acquisition method that does not rely on the Mac App Store.

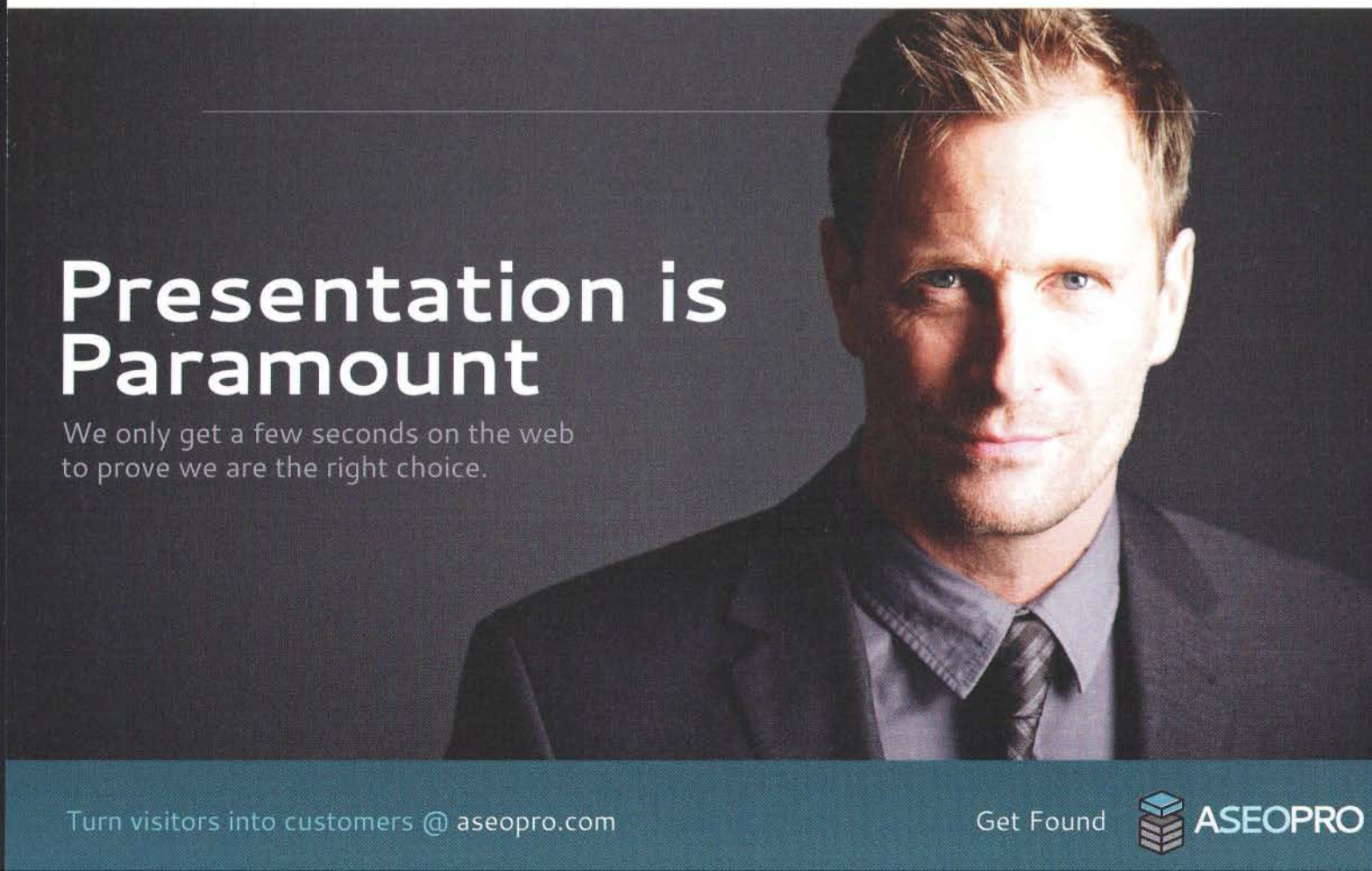
Installation

Once an organization has purchased your application and enough user licenses, they will want to install it on (possibly) a large number of machines. Large organizations will generally not have armies walking around and manually running your installer by double-clicking it. Instead, they will be using a software deployment mechanism that automatically installs software on large numbers of machines. Your software's installation method must work with enterprise software deployment systems.

Some automated software deployment tools for Mac OS X commonly in use include: Apple Remote Desktop, JAMF Casper, Absolute Manage, Puppet and Munki.

This list of deployment tools may have you wondering: "How can I make sure my software can be installed by all these tools? That sounds like it will be difficult and time-consuming to test!" But in truth, it's not that hard as long as you adhere to some basic principals, since most automated software deployment tools use similar mechanisms for the actual installation.

First: use the Apple package format. It's not perfect, but every major software deployment mechanism works with Apple's packages. Apple itself leverages the Apple package format to install software suites like iLife and Final Cut Studio, and even to install Mac OS X itself. So it seems reasonable that it is possible to install *your* software using Apple's package format. If, instead, you create your own installer



application or use a third-party installer technology like InstallEase or InstallAnywhere, it's likely your installer will not work in an enterprise environment.

Second: avoid pre- and post-install scripts if at all possible. If you must utilize scripts in your package, test them in scenarios found in enterprise deployments. We'll go into more detail about that in a bit.

Third: your package should install everything needed to run your software. It should not leave additional installation tasks for the first launch of your software. Remember that in an enterprise environment, the person who first launches your software after install may not have administrative privileges. They will be annoyed when your software asks for administrative credentials the first time they try to use it. Your software's installer has administrative privileges – it should do things that require admin rights at that time.

The "fancier" your installer, the greater the chance it will do something that makes it incompatible with an enterprise install. So keep your packages simple and test them for unattended installation.

Testing your Package

The simplest and most important thing you can do as a software developer to ensure your software's installer package is enterprise-friendly is to use an `ssh` session and the command-line installer tool (`/usr/sbin/installer`) to install your software on a machine. Test the install both with no one logged in as a GUI user and also with a GUI user logged in. Here's an example:

```
% ssh gneagle@aquaman
Last login: Tue Mar 29 10:56:17 2011
gneagle% sudo installer -pkg /tmp/foo.pkg -target /
installer: Package name is Foo
installer: Installing at base path /
installer: The installation was successful.
```

Here we install `Foo.pkg` using the command-line `installer` on the machine "aquaman". Even though the installation completed successfully, we must manually test the software to be sure it functions as expected. You should perform the command-line installation test with no user logged into the machine (with the login window displayed), as well as with a user logged in to make sure the installation completes correctly in both scenarios. When software is deployed in an enterprise, any given machine may or may not have an active user logged in, so you must test with no logged-in user. Be sure to test under Tiger, Leopard and Snow Leopard if your application runs under all of those versions of Mac OS X.

If your software can be installed successfully with this method and your software functions as expected when installed this way: congratulations! There's a very good chance your installer package is enterprise-friendly, and will work with all the major software deployment systems.

FontAgent Pro
Professional Font Management

Designed for IT Managers and Creative Pros

"FontAgent® Pro is the premier font manager for creative Mac users, giving them the power and simplicity to get projects done fast and right. It's the right technical solution as well. The new FontAgent Pro™ TeamServer™ is perfect for small workgroups, and the Enterprise Server provides slick directory integration and unrivaled scalability."

"As integrators and consultants, we have been recommending FontAgent Pro to our clients for the last seven years because it delivers unparalleled functionality at a competitive price backed by great, friendly service. You can't ask for more than that."

Joe Schram and Jim Hollis
Sparktivity
Raleigh, North Carolina

Insider™

800-866-8778 +1-408-871-9933
www.insidersoftware.com

© 2010-2011 Insider Software Inc. All rights reserved. FontAgent is a registered trademark and FontAgent Pro, TeamServer and the Insider logo are trademarks of Insider.

iTraining Bootcamp
Get Apple Certified.

Produced By: Future Media Concepts

Authorized Training Center

Introducing **iTraining Bootcamp**
a unique, intensive training & certification event for Apple users

Get Certified:

- Mac OS X Snow Leopard
- Mac OS X Snow Leopard Server
- Xsan 2
- Final Cut Pro Levels 1 & 2
- Final Cut Server
- Color
- Motion
- Logic Levels 1 & 2
- Sound Editing in Final Cut Studio
- Aperture

May 17-24, Orlando, FL
Airfare*, hotel & meals included with your registration!
*Up to $300

www.iTrainingBootcamp.com **877-362-8724**

Pre- and Post-install Scripts

The most common thing that causes a package to fail to be enterprise-friendly is poorly written pre- or post-installation scripts included as part of the package.

If your installer makes use of pre- and/or post-install scripts, test them during your remote install tests to make sure they don't do anything visible. They should not open Finder windows or launch your application or quit other applications or modify the current user's Dock. Even if you think actions like these are desirable when your package is installed interactively, they are definitely undesirable when a system administrator is trying to install your software to hundreds or thousands of Macs.

If you really must have your installation interact with the user when doing a manual install, you can still write your scripts to do the right thing during an unattended install. A pre- or post- script can tell if it's being run in the context of a non-GUI install: the `installer` command sets the COMMAND_LINE_INSTALL environment variable. Just test for it and skip over a task that's inappropriate when installing at the command line. Here's a Perl example from a postflight script in the iTunes install package that updates the Dock:

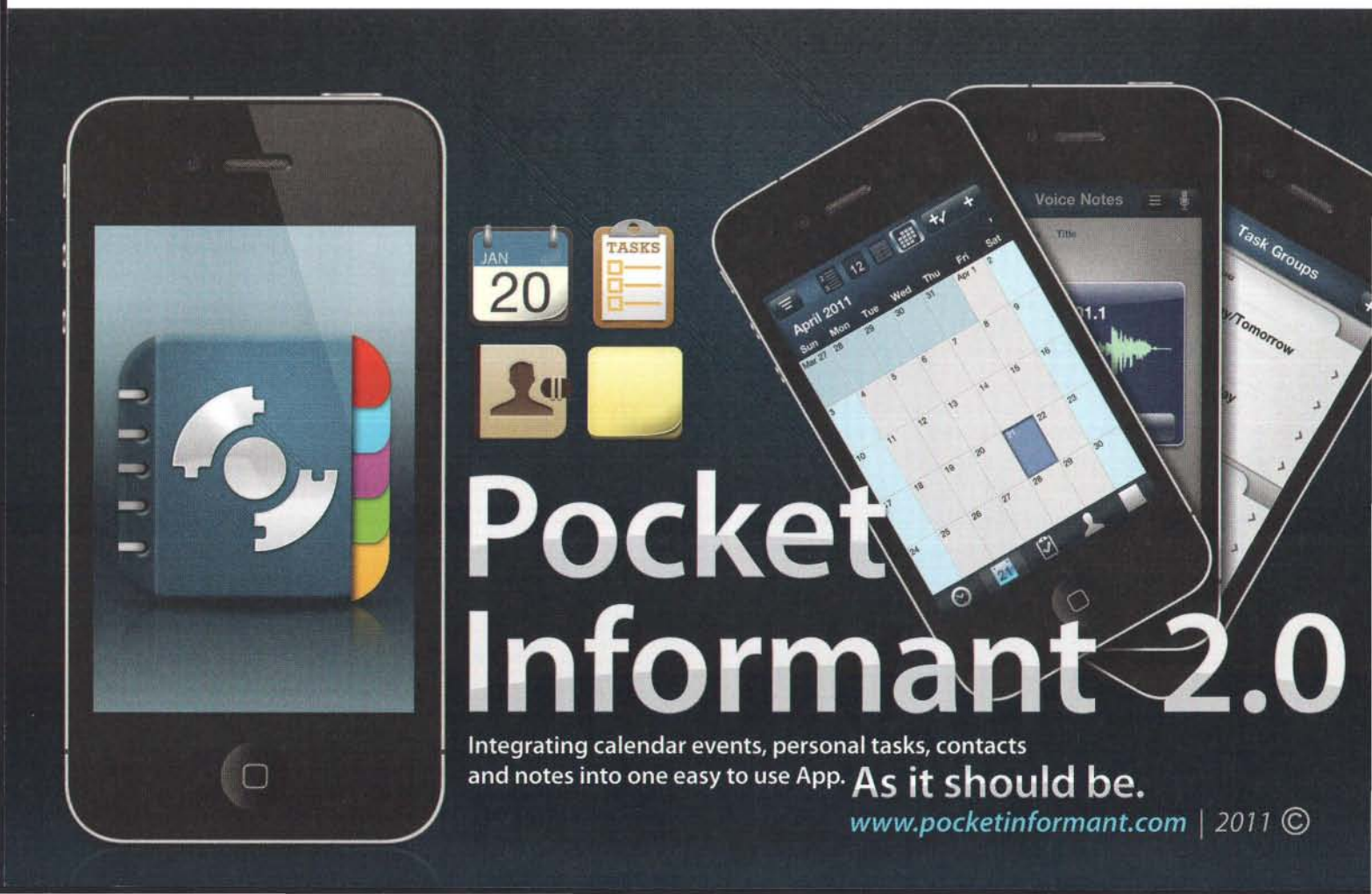
```
# exit if command-line install
exit(0) if ($ENV{'COMMAND_LINE_INSTALL'});
# update user's Dock
...
```

If the package is installed interactively, the Dock is updated for the current user. But if it's installed via the command line, the script exits, leaving the Dock untouched.

Another common mistake in pre- and post-install scripts is the use of the $USER variable to attempt to get some information about the currently logged in user, or to access the user's home directory. When the installer is run at the loginwindow by the root user, "$USER" is undefined and scripts that use that variable may not perform as expected.

Installer packaging extra credit

There's an additional scenario your installer package might encounter in an enterprise deployment. A common practice in large organizations is to configure new machines using an "image". Installation images capture the entire state of a startup disk: the OS, installed software, and the configuration of each. There are a couple of methods of building installation images that use Apple packages to install the OS and all extra software and configuration files. The unusual bit is that when these images are built, the installer packages are used to install software on a disk that is not the startup disk. Some installer packages make assumptions in pre- and post-install scripts that they are installing only to the startup disk and fail to run correctly when the installer applies the package to a disk other than the startup disk. For extra credit and better compatibility with this workflow, test the installation of your software on a disk other than the startup volume.

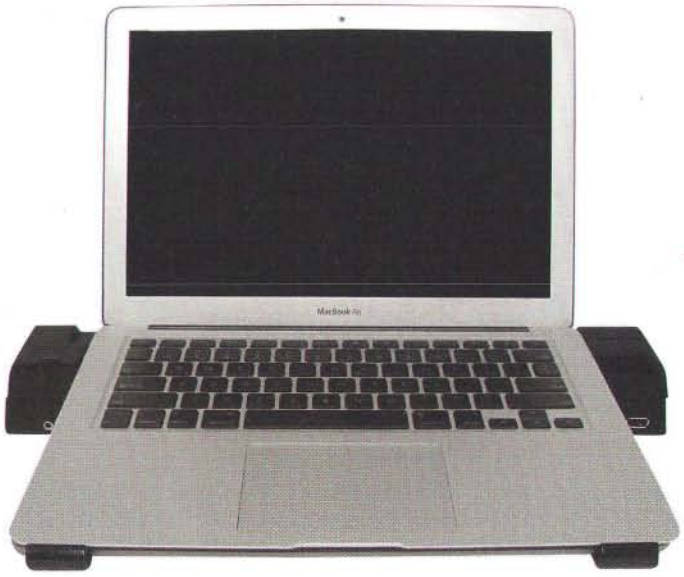


BookEndz®

Docking Stations for Apple Computers

DOCKING STATIONS FOR APPLE COMPUTERS

Turns your laptop into a desktop



NEW!
MacBook Air
11" and 13"
Docking stations

- Includes USB hub to give you a total of 4 USB 2.0 ports
- Allows access to SD slot
- Adds an Ethernet Connection to your MacBook Air for wired networks
- Allows you to use an external monitor in addition to your laptop
- Angles your MacBook Air off a flat surface

Visit our website for latest product announcement www.BookEndzdocks.com

BookEndz®

Manufactured by Olympic Controls
1250 Crispin Drive • Elgin, Illinois 60123 • USA

Phone: 847-742-3566 • Fax: 847-742-5686 • Toll Free: 888-622-1199 • E-mail: Sales@BookEndzdocks.com

Post Installation

You've made it easy for an enterprise to purchase and install your software. That's a good start. But there are post-installation issues that can make your software easy or hard to deal with in an enterprise environment.

Activation, Registration and Licensing

Much commercial software requires some sort of activation, licensing, and/or registration before it can be used. If activating or licensing your software requires manually typing in a long code on first launch, it's not enterprise-friendly. Consider providing a method for enterprise licensing and activation: perhaps a licensing package that can be installed, or at the bare minimum a command-line tool that can be run. Some organizations have implemented network licensing servers or software asset management systems; consider supporting these.

User registration, in which you capture some contact information about the user of your software, may make a lot of sense for individual purchases. But in an enterprise deployment, is it really valuable to get incomplete and inaccurate info from hundreds of employees of the same company? Presumably you already captured the important information when you sold the multi-user licenses for your product to that company. Consider providing a way for a systems administrator to pre-register software before deployment and/or turn off any user registration prompts. This mechanism should take effect for all users on a machine; it's not helpful if it requires modifying something for each user.

Remember also that the person using a machine day-to-day may not be the person who installed or activated the software. A technician who must manually install or activate your software may do so while logged in via his or her account, or while logged in via a local administrative account. If that activation is stored somewhere in the home directory of the user who did the activation, it won't be available to the "real" user of that machine when he or she logs in next. This problem is even worse in the education arena, where there may be many users sharing a group of machines. Installation and activation of your software should make the software available and usable by all users of a given machine. This means storing the activation/licensing info somewhere readable by all users. `/Library/Preferences` or `/Library/Application Support` are good candidate locations for licensing/registration information.

Additional First-Run Installs

Several commercial software packages offer to install additional components on the first launch after installation, prompting for administrative credentials. This is problematic in an enterprise environment, as the user running the software for the first time may not have admin rights. You can avoid this issue by making the additional components part of the original installation package, possibly as optional installs. If this isn't possible, provide a manageable preference to turn off the installation of optional components. See the discussion on managed preferences later in this column.

Even if your software offers to install items in the user's home directory (and therefore doesn't need administrative credentials), consider the scenario where there are multiple users on a single machine – does your software really need to copy the same files to every user account? If these files are templates, examples, or stationery, consider making them available to all users from a shared location, and install them there as part of the installer package.

Updates

Many vendors have their applications check for their own updates. Some use the popular Sparkle framework for this functionality; others roll their own solution. This is a great strategy for individual purchasers like home users, where the purchaser is the primary user and essentially the administrator for his or her own machine. But again, in an enterprise environment, applications that check for updates can be an annoyance. Bandwidth is wasted when one thousand copies of an application, all installed in a single company, each go out to the internet and retrieve one thousand copies of the update. Worse, once they've downloaded an update, they might alert the user of the software and ask for administrative credentials that the user doesn't have.

For these reasons, it's essential that you provide a way for system administrators to turn off any auto-update mechanism for your software. You must also make updates available via an alternate mechanism that can be installed the same way the original software was installed. That means updates that are distributed as standard Apple packages. The mechanism for disabling update checks should be global – that is, it must work for all users of a given machine. That could take the form of support for Apple's managed preferences framework, or just a preference file in a globally accessible location. Be sure to document this!

Preference Management

Since we mentioned Apple's managed preferences framework, this is a good opportunity to talk about preference management. In large organizations, it is sometimes helpful for system administrators to be able to manage settings or preferences for a piece of software. Sometimes this takes the form of setting helpful initial defaults, or turning off inapplicable or unsupported functions. Other times this might involve setting and enforcing an organizational policy.

You can at the very least make your software's preferences manageable at the most basic level by storing your software's preferences in Apple's property list format in the standard locations (`~/Library/Preferences` and/or `/Library/Preferences`). Preferences stored this way are manageable via Apple's managed preferences frameworks. If you use CFPreferences from the CoreFoundation framework

Don't get Lost in the Crowd...

...let an Evolis card printer Identify you!

Fully compatible with Mac®, the Pebble[4] ID card printer is the perfect solution to print all your ID cards in high-resolution. Whether you need Employee ID's, Student ID's or Loyalty cards, the Pebble ID card printer will help you stand out from the crowd.

Call TransTech Systems today

1-888-843-3643

to get connected with your local Evolis partner!

Pebble[4]
Color Single-sided card printer

TRANSTECH SYSTEMS
IDENTIFYING SOLUTIONS

TransTech is an Official Evolis Solutions Provider
Tel: 1.888.843.3643 • **Fax:** 503.682.0166 • **email:** sales@ttsys.com • **www.ttsys.com**

EVOLIS OFFICIAL PARTNER

Mac is a registered trademark of Apple Inc.

or NSUserDefaults from the Foundation framework to handle your software's preferences, you'll have even better compatibility with Apple's managed preferences.

Et cetera

Here are a few other things that might differ between individual home/small business users and users in an enterprise environment:

Enterprise users may have network home directories instead of home directories on the local disk under `/Users`.

Even without network home directories, enterprise users are more likely than home or small business users to store data on file servers.

Don't assume file servers are AFP (Apple File Protocol) servers; in large organizations Windows (SMB/CIFS) file servers are the rule, and NFS file servers aren't unknown.

Enterprise laptop users are more likely to be using FileVault to protect their data.

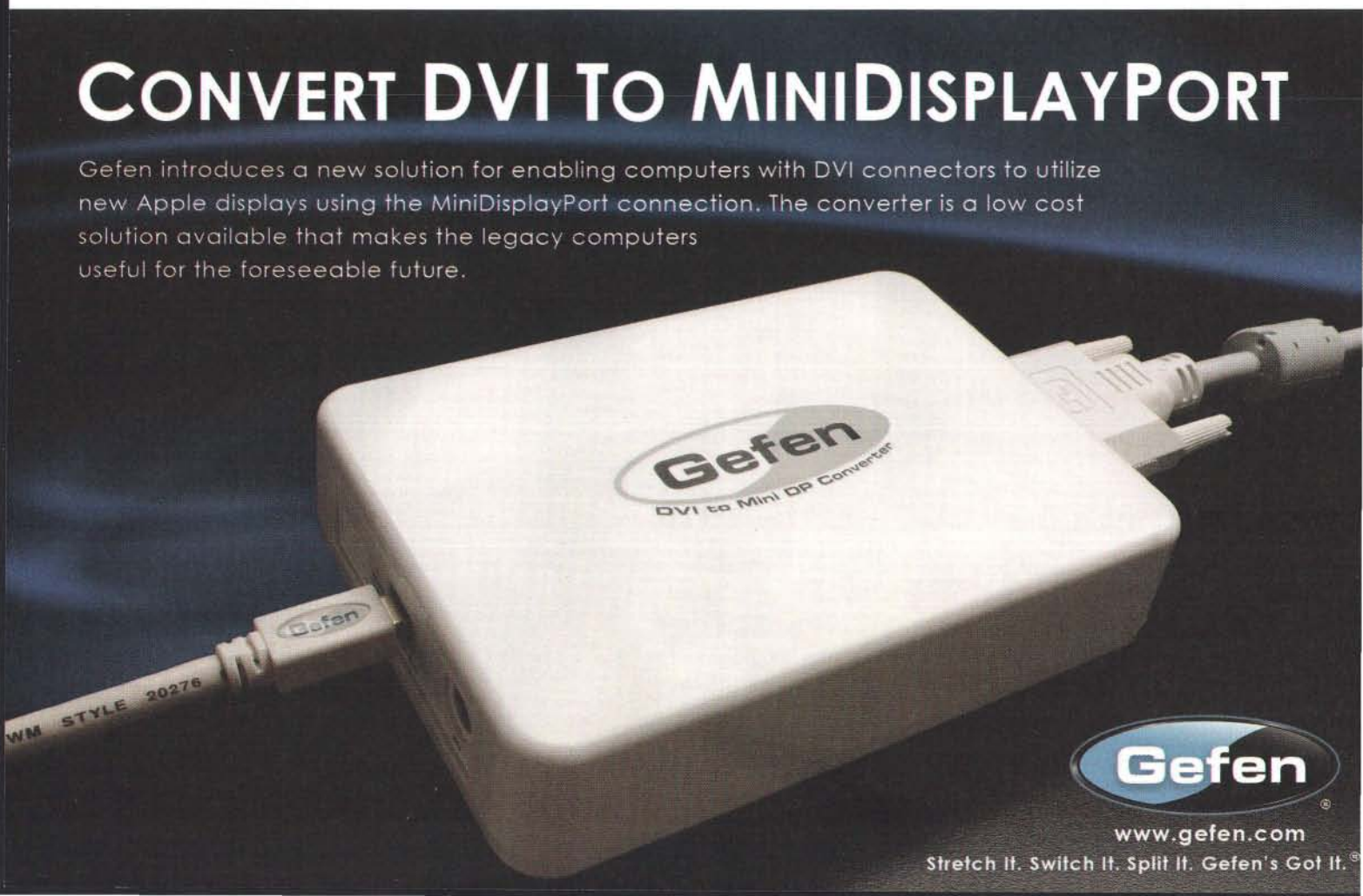
If your software isn't tripped up by any of these, it's more likely to play well in the enterprise.

Conclusion

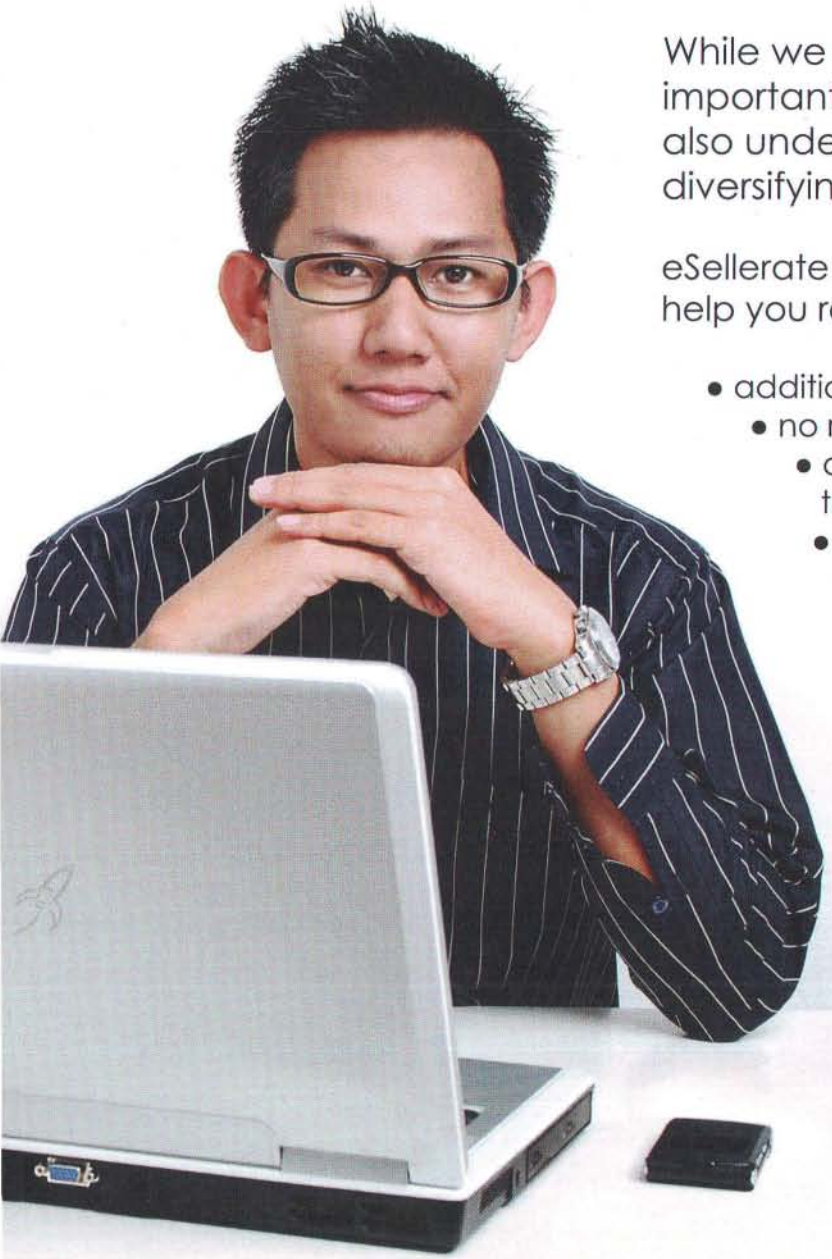
Making your software enterprise-friendly need not be difficult. A little planning and consideration of enterprise issues will help you avoid common pitfalls. Package your software using the Apple package format and test command-line installations. Provide enterprise administrators with ways to license and/or activate your software via the command line, by installing a package containing licensing information, or by using a network license manager or asset manager. Give enterprise administrators the ability to suppress registration dialogs and auto-updates. Store your software's preferences and configuration in the standard Apple property list format, or even better, use Apple's preference frameworks. Remember that in a large organization, your software's users may not be administrators, and that an enterprise machine may have more than one user. Follow these guidelines and you'll have software that's easy for enterprises to buy, install, configure and use. Who wouldn't want to sell a few hundred thousand extra copies of their software?

About The Author

Greg Neagle is a member of MacEnterprise (macenterprise.org) and is a senior systems engineer at a large animation studio. Greg has been working with the Mac since 1984, and with Mac OS X since its release. Greg Neagle and Edward Marzcak's book:* Enterprise Mac Managed Preferences, *which covers Apple's Managed Preferences, was recently published by Apress. Greg can be reached at gregneagle@mac.com.



BEYOND THE APP STORE
building a complete sales solution



While we agree that an app store is a very important part of your sales strategy, we also understand the importance of diversifying your methods.

eSellerate provides you with several options to help you round out your plan:

- additional distribution channels
- no restrictions on third-party APIs
- access to customer data allowing you to market directly to them
- ability to offer purchase options from within trial or demo applications
- ability to offer updates or upgrades from within your own application
- absolute control over your pricing with no limitations
- maintain complete control of your products and strategies
- no need to maintain multiple code sets
- rates as low as 5.9%

esellerate®
Digital River™

Supporting great Mac Publishers since 2000.
Competitive rates for all businesses, starting at 5.9%.

www.esellerate.net/mactech

+1 402-323-6600 x39701

CORESEC: SECURITY TOPICS FOR ADMINISTRATORS AND PROGRAMMERS

Secure Shell (SSH)-A Hidden Gem, Part 2

SSH is a powerful tool with many uses. Let's explore this staple of OS X

By Michele (Mike) Hjörleifsson

Introduction

Last month we looked at the basic underpinnings of SSH: how to configure the client configuration files, server configuration files and create key based authentication. We also reviewed some of the built in utilities and applications that come with OpenSSH. This month we will dive a bit deeper, looking at SSH as a communication tool for other applications to use when a secure, simple connection needs to be made from one host to another to accomplish a variety of tasks.

Command Line SSH Goodness

Last month we looked at scp as a method for performing a secure copy from or to a remote host, including the recursion option to mirror a directory from one host to or from the other. While this is great for a single operation, it doesn't lend itself to the backup and recovery operations that administrators perform all the time. Rsync, from the people that brought us SAMBA (windows SMB integration), allows you to do fast incremental transfers/backups of data over SSH locally or across the globe. For example, using the command below, rsync would look at the /backup/Users directory on your local machine, compare it to the content of /Users on the remote host and copy any files that needed copying from the remote host to the local machine all over a nicely secured SSH session. The benefits to this versus sftp or scp are twofold; first, you aren't making a full copy of files you already have on the local side, and second, it's therefore much faster.

```
rsync -avz -e ssh root@remotehost.com:/Users/* /backup/Users
```

The command line options –avz tell rsync that this is an archive (a), to perform the operations in verbose mode (v) so we can see what is happening and to compress the data traffic (z). The –e option tells rsync to use SSH as its communication protocol instead of it's own client/server protocol (which would require you to setup an rsync launch daemon on your server.)

For this to work in a script that runs unattended, you would need to create ssh keys and setup the authorized_keys file on the remote host (see last months article) and ensure that the user you are using to connect with has permissions to read the files / directories in question.

Mounting Remote Filesystems

Another wonderful command line set of tools are the combination of MacFuse and SSHFS from Google. You can download MacFuse from http://code.google.com/p/macfuse/downloads/list and install on the machine you would like to connect from. MacFuse provides the File System in User Space (FUSE) environment originally developed for Linux. The FUSE system allows users to load and unload file system drivers on the fly rather than embedding them in the kernel or a kext file. SSHFS on the other hand is a pluggable module for FUSE that allows you to mount a remote file system over SSH as if it were local. Installing SSHFS is a little bit more complex and is currently only working in 32-bit mode (vs. 64 bit), though it works well. To install SSHFS, you need to download the binaries from http://osxbook.com/download/sshfs/sshfs-static-leopard.gz or via svn

```
svn co http://macfuse.googlecode.com/svn/trunk/
filesystems/sshfs/binary sshfs-binaries
```

Once downloaded you need to copy the sshfs-static-leopard to a location in your path, for instance:

```
sudo cp ./sshfs-static-leopard /usr/sbin/sshfs
sudo chmod 755 /usr/sbin/sshfs
```

Once in the path and executable you can connect to remote hosts using the a simple syntax:

```
sshfs user@host:/path /mypath -
oauto_cache,reconnect,volname=<volname>
```

You will need to ensure that /mypath is a valid directory on the local filesystem. If you have setup SSH keys, it will just connect. If not, you will be prompted for the password for the remote system. Once connected, the remote filesystem can be seen in Finder or a shell and treated like any other network share.

Start a Web Site Today!

from only $4.95/month

Get first month for a penny!
Coupon code: **MACTECH**

Powered by 130% Texas Wind Energy!

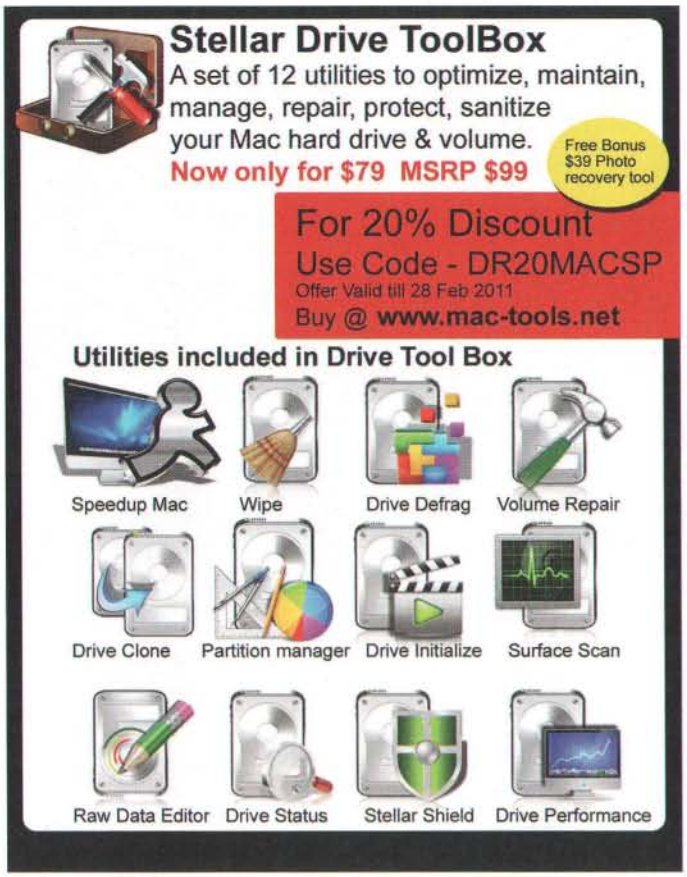
100% Mac-compatible Web Hosting

- 99.9% Uptime
- 45-day Money Back Guarantee
- Over 4,500 Free Web Design Templates
- Free Site Builder Software
- In-house Support Available 24x7x365

Sign up at www.HostGator.com/mactech and get the first month for just a penny!

Already have a web site? We'll transfer it for free!

www.HostGator.com 866.96.GATOR



Protecting SSH

One last command line gem is DenyHosts, an open source project that tracks SSH logon attempts and stuffs potential hackers into your hosts.deny file automatically preventing them from trying to connect again. You can use this tool on Mac OS X or Linux and its freely available for download at http://sourceforge.net/projects/denyhosts/

Graphical Tools

The graphical side of the SSH picture is even prettier than the command line tools, and I don't just mean the look and feel. Several developers have made some wonderful SSH tools that are either Freeware, Shareware or paid for software. File transfer software like Cyberduck, Transmit, Filezilla, Interarchy and most FTP graphical applications support SFTP. This provides the ability to securely move files between your systems with a nice graphical interface. Super Flexible File Syncronizer from Super Flexible Software (http://www.superflexible.com/features.htm) enables you to do graphical directory synchronization over SSH and other Internet protocols.

YazSoft provides a utility called ShareTool 2, which allows you to see and connect to Bonjour advertised resources from anywhere in the world securely over SSH. For instance if you are in a hotel in Vegas you can access your shared iTunes library back in New York and play some tunes or even watch a video (if the bandwidth at the hotel can handle it.) Additionally they have written hooks so the Bonjour advertised resources show up in their respective applications as if you were still local to those resources.

Virtual Private Networks & Remote Screen Sharing

Remote access to multiple resources like a being able to screen share with any machine on a remote network (versus just one that you redirect a firewall port to) or being able to access any resource on that network as if you were in the office requires some type of secure remote connectivity, like the VPN provided by OS X. But using L2TP is cumbersome, especially at the remote end. Oftentimes, the required ports aren't open and if you are at someone else's facility or a hotel, for instance, then you won't have the ability to open those ports, enter SSH based tools as a solution.

Simple tunneling can be achieved with graphical software like and SSH Tunnel Manager (Freeware available for download at http://projects.tynsoe.org/en/stm/). Another application called Meerkat (Shareware) from Code Sorcery Workshop is one of my favorite SSH utilities. Why? Well Meerkat helps you to create SSH tunnels to remote hosts that send traffic from your local machine to the remote network via SSH. Though you can do this from the command line, Meerkat allows you to setup connections, and even triggers that start the connections automatically when you launch an application. Oh, and it is totally scriptable, reconnects disconnected sessions and supports SSH keys or keychain stored passwords.

Another freeware application called Secret Socks by an independent programmer (Joshua Chan) in Canada, allows you to setup a SOCKS proxy over SSH, providing remote connectivity to websites, ftp etc, from the proxy for any application that support SOCKS proxies (like Safari.)

ChickenofTheVNC is a freeware utility that will allow you to gain access like Screen Sharing does to a remote graphical console on any platform that supports VNC connections (OS X, Windows, Linux, Solaris etc.) and has the capability of tunneling VNC over SSH for security and port simplicity (all traffic runs over SSH.)

OpenVPN and TunnelBlick provide an server and client combination for establishing an SSH based VPN (virtual private network.) OpenVPN can be installed from MacPorts on your server, though it will require some configuration file typing to get going (here is a decent how-to http://tinyapps.org/docs/openvpn/) but is very flexible. Once up you can install Tunnelblick on your client machines, which will create the virtual adapter and provide a nice graphical setup to connect to your OpenVPN server.

One question I get often when teaching Support or Server Essentials is how an administrator can provide support via Screen Sharing to a set of Apple computers on a remote network. There are two methods I see used often and one I tested and started telling my students and clients about. You can use iChat and its associated screen sharing capabilities, or you can setup Apple Remote Desktop (ARD) on a machine inside the network and open the ARD ports to that machine so you can remote in, then remote to machines on the network. These are the two graphical methods. You can also punch a bunch of holes in the firewall and have each of the Macs answer Screen Sharing on a different port by changin their config files, or you can use an SSH tunnel to connect into the network then onto a host. None of these methods are optimal in corporate environments. iChat turns on the microphone and increases bandwidth by sending audio, since you are typically already on the phone with the person requesting support this is silly. Additionally, there is no way to turn down the color depth from the graphical interface to reduce bandwidth requirements. ARD via ARD is a challenge as you are screen sharing another screen sharing session. Tunneling is fine, but you need a tunnel entry for each host on the remote network. So I did some research into a tool I have used often in remote Linux support environments called NoMachine or NX.

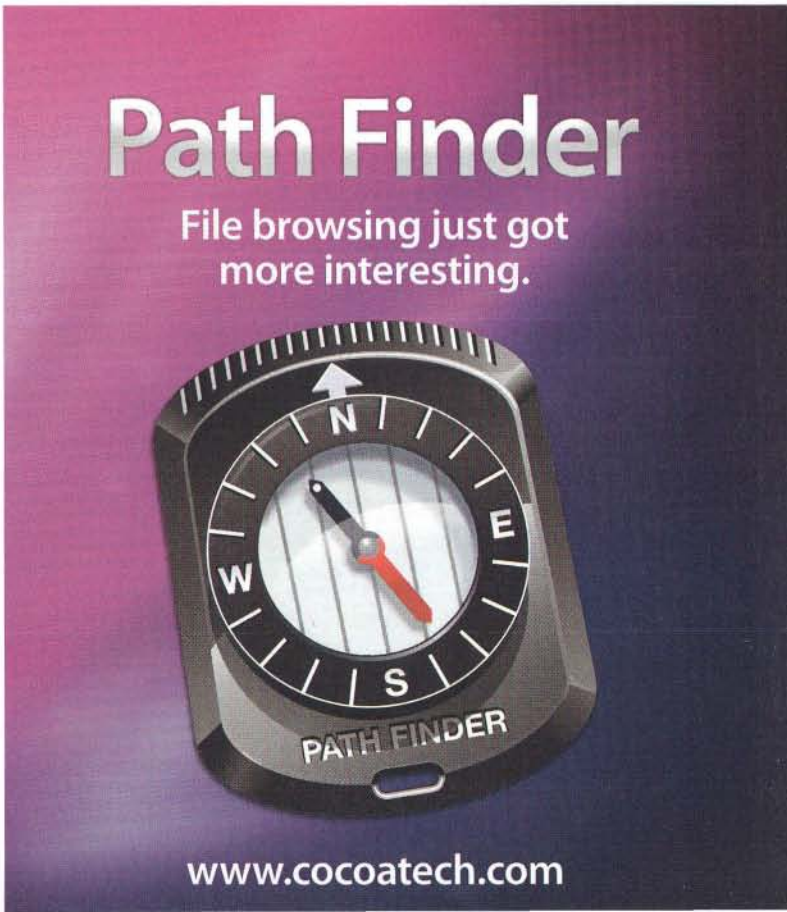
NoMachine is another neat open source set of tools that allow you to remotely screen share all the machines on your internal network through a single entry point to your network. Currently there is no Mac OS X implementation of their server software though it is in development. Why let that stop you? I installed NoMachine in a VMWare Fusion guest on the server and with a little configuration am able to use their OS X client software to connect to the NX server and through to all the machines in my office to provide fast remote screen sharing without having to do a ton of firewall tricks. You will need to setup a VNC password and enable the Screen Sharing and Remote Login on each machine you want to connect to, configure them as targets in NX and voila, instant support to an entire network over SSH. Additionally you can connect to Windows machines and even Linux machines using the same NX server and OS X client software. (Note: X11 is required on the client machine you connect from)

Conclusion

SSH has been around since 1995 and is still one of the fastest and most secure facilities for connecting to remote hosts and networks without a ton of effort or sacrificing security. With developers utilizing the OpenSSH framework to add additional features or tunnel their applications traffic over SSH, SSH and the associated tools that use it will be around for a long time and as an administrator you should familiarize yourself with some of these wonderful little gems.

About The Author

Michele (Mike) Hjörleifsson, co-author of the* Apple Training Series: Security and Mobility *courseware has been developing on the Apple platforms since the Apple][+, implementing network and remote access security technologies since the early '90s, and worked with the nation's largest corporations and government institutions, authoring white-papers, technical magazine articles and topical discussions at IETF (Internet Engineering Task Force), and other organizations on security topics, and podcasting with Apple Podcast Producer. He is currently working with companies worldwide on Apple and Security consulting projects and conducting Apple IT and Pro Apps training. Feel free to contact him at mhjorleifsson@me.com



Meet Cappuccino

Use your Cocoa skills to build stunning web applications

by Johannes Fahrenkrug

Introduction

Web development can be a challenge. You need to learn multiple languages and technologies, like HTML, CSS and JavaScript. To build truly impressive applications, you even have to dive pretty deep into these technologies and master them. Wouldn't it be nice if you could use your long-time knowledge of Cocoa that you have gained and perfected over the years to build web applications? That's exactly what Cappuccino makes possible.

What is Cappuccino?

Cappuccino is a framework for developing desktop class web applications. It consists of 2 important parts: the language Objective-J – a re-implementation of Objective-C in JavaScript – and the Cappuccino libraries – a Cocoa-clone written in Objective-J (hence the dark, hot beverage inspired name). This combination of language and libraries runs in modern browsers without any plug-ins and creates to necessary HTML, JavaScript and CSS to execute your application. Just like a desktop application, Cappuccino only covers the client side of your development: Since the application is running in the browser, in most cases you will still need to use something like Ruby on Rails, Django, or PHP to build a server component that your Cappuccino application can communicate with to load and save data.

All this might still seem kind of abstract to you. To really understand what a powerful piece of technology Cappuccino really is, the next paragraphs will show you how to install the tools and how to build your first Cappuccino app!

Because Cappuccino is based on Objective-C, you should be familiar with Objective-C and the tools that surround it (Xcode, Interface Builder and so on). This article assumes that base knowledge.

Installation

To build the Cappuccino tools, you must have `gcc` installed. Since it is part of Xcode, you should already have it installed. Next, you need to check out the Cappuccino git repository or download it as a zip file from http://github.com/280north/cappuccino/zipball/master. Unpack the zip file, open `Terminal.app`, change into the folder you just unpacked, and run the `./bootstrap.sh` script (you might need to run it as `sudo`, depending on your system setup). This will download and install all the required tools. You can safely accept all the default settings with `yes` or by pressing return.

As a final step, you need to add the new binaries to your path. Do this by appending `export PATH="/usr/local/narwhal/bin:$PATH"` to either your `~/.profile` (bash) or `~/.zshrc` (zsh) file. To activate these new settings either open a new shell or run `source ~/.profile` (or `~/.zshrc`). Now you're all set to build the actual Cappuccino libraries: back in `Terminal.app`, run `jake sudo-install`. That will take a little bit.

Hello World

A new Cappuccino application is created with the `capp` command-line tool which creates the basic directory layout with the necessary files and frameworks. The application we are going to build will be called `image_search`. Run this command to create it:

```
capp gen -t NibApplication image_search
```

The `-t NibApplication` switch tells the `capp` tool that you want a use a template for an application that has a user interface which you can edit in Xcode. Cappuccino cannot directly read `nib` or `xib` files. They first have to be converted into a `cib` file with the `nib2cib` tool. To do that, change into the `image_search` directory and run `nib2cib Resources/MainMenu.xib`. When you open `index.html` in the `image_search` directory in Safari after that, you should be greeted with Figure 1. When you move the slider

back and forth the value in the text field should change: You're first Cappuccino application works!

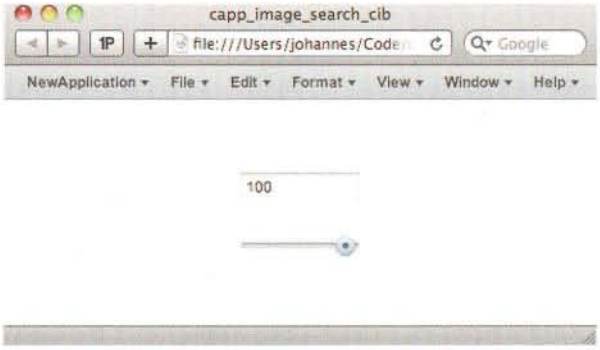


Figure 1. Nib-based, unedited Cappuccino application

Making Changes with Xcode

Maybe you're not quite convinced yet that the UI from `MainMenu.xib` is really being used. To erase those doubts, we will edit the UI in Xcode. Since Xcode 4 doesn't allow you to define custom outlets and actions directly in Interface Builder anymore, the Cappuccino team has developed a tool called `xcodecapp` that sets up a dummy Xcode project for you to be able to easily edit your xib files. To use it, simply run `xcodecapp` inside the `image_search` directory without any arguments. You'll now change the application so that you can enter your name and be greeted when you click on a button. In Xcode (or a different editor), open `AppController.j` and edit it so that it ends up looking like Listing 1.

Listing 1: AppController.j

```
@import <Foundation/CPObject.j>

@implementation AppController : CPObject
{
    @outlet CPWindow    theWindow;
    @outlet CPTextField textField;
}

- (void)awakeFromCib
{
    [theWindow setFullBridge:YES];
}

- (IBAction)sayHi:(id)sender
{
    alert("Hi, " + [textField stringValue]);
}

@end
```

The `xcodecapp` tool automatically picks up those changes so that we can use the new outlets and actions in Interface Builder. Next, click on `MainMenu.xib`, select `Window`, and replace the slider with a button and set its autosizing attributes as shown in Figure 2. Now Ctrl-drag from the AppController object to the text field and connect it the `textField` outlet. Then Ctrl-drag from the button to the AppController object and connect it to the `sayHi` action. Save it.

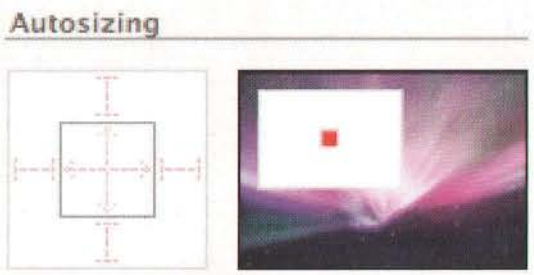


Figure 2. Autosizing attributes for the button

When you reload index.html in your browser now, you'll see the text field and the button. Enter your name and press the button and you'll be warmly greeted by Cappuccino! Congratulations: You have written your first own Cappuccino application, complete with an action and an outlet.

You see that Cappuccino development is very similar to the Cocoa development you're used to. That's why we don't want to waste any time: A more complex app awaits!

A More Complex Application

The application that you will build next is a client for the Google Image Search API: Results will be displayed in a table on the left side of the screen and when a search result is selected, the image will be displayed in full size on the right side. (While typing this yourself is good practice and gets you more familiar with the tools, the source for this final application can be found at ttps://github.com/jfahrenkrug/CappuccinoCibImageSearch.)

Go back to Xcode and remove the button and the text field from the interface. Drag an `NSSplitView` to the window and size and position it to fill the whole window and set its autosizing attributes as shown in Figure 3.

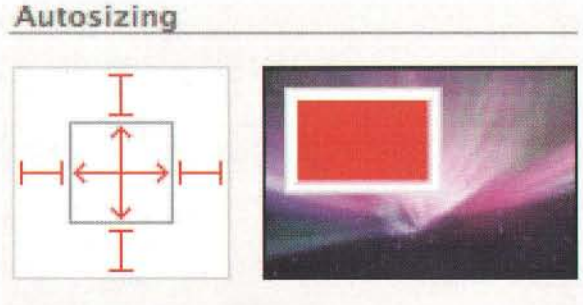


Figure 3. Autosizing attributes for the split view, table view and image view

STOP SHARING!

START FAXING!

Each subscriber receives faxes directly by email as PDF file attachments.

Corporate accounts from 3 to 100+ users available

For more information and a special offer for MacTech readers, visit

www.MaxEmail.com/MacTech

maxemail®

Call: 800-964-2793

Then add a text field (autosizing Figure 4) to the top left area of the split view and a table view (autosizing Figure 3) right under it.

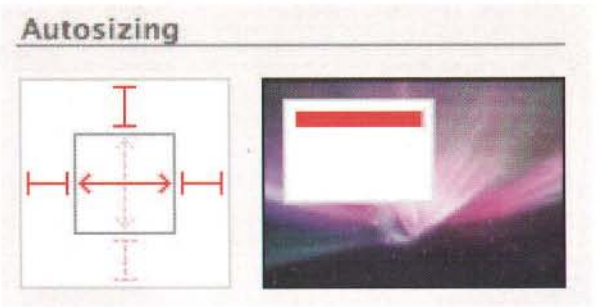


Figure 4. Autosizing attributes for the text field

Name the left column of the table "Title" and the right one "Size." In the inspector for the left table column set its identifier to `title`. Set it to `size` for the right column. Those identifiers are needed later for the `CPTableViewDataSource` delegate methods to determine which column to return data for. To be able to display the image, add an image view to the right half of the split view, setting its autosizing attributes according to Figure 3. The interface should look like Figure 5 by now.

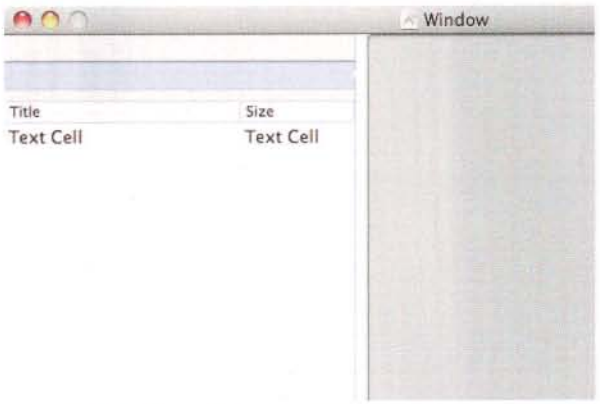


Figure 5. The finished UI of the image search application

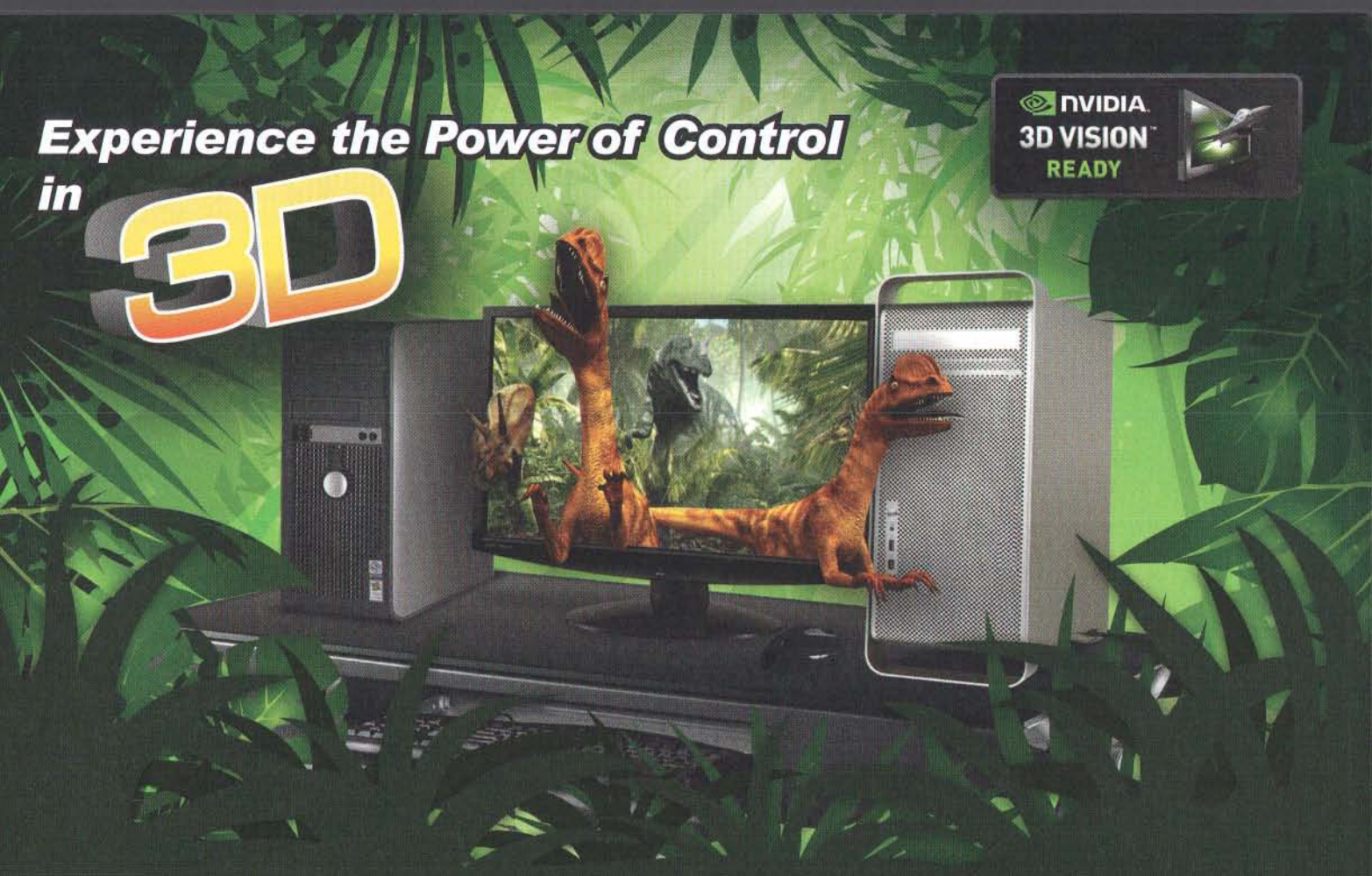
Now we just have to create and connect the outlets and actions. To do that, we first have to define them in the `AppController.j` file as shown in Listing 2 (two extra outlets, two instance variables that we will need later on, and one empty action).

Listing 2: AppController Outlets, Actions, and Instance Variables

```
@implementation AppController : CPObject
{
    @outlet CPWindow    theWindow;
    @outlet CPTextField textField;
    @outlet CPTableView tableView;
    @outlet CPImageView imageView;
    CPArray images;
```

Experience the Power of Control in 3D

NVIDIA 3D VISION READY



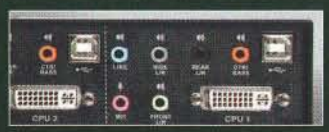
IOGEAR 2-port Dual-Link DVI KVMP Pro Switch with 7.1 Audio

This powerful KVMP supports 3D graphics with NVIDIA® 3D VISION, is compatible with gaming keyboards and has an integrated 2-port USB 2.0 hub for peripheral sharing. Now users can control two computers and share multiple USB peripherals from a single keyboard, monitor, and mouse console. Increase efficiency and save equipment cost for applications where ultra-high video resolution, large or widescreen monitors are required such as:

- CAD (Computer Aided Design)
- CAM (Computer Aided Manufacturing)
- Medical / Surgical / Diagnostic Displays
- Desktop Publishing / Graphic Design
- Video Editing
- High-End Gaming

Share a professional quality monitor, keyboard, mouse and 7.1 surround sound system between two USB / DVI computers

Supports high-end 3D graphics with maximum video resolutions of:
- DVI Dual link: 2560 x 1600
- DVI Single link: 1920 x 1200



DynaSync* technology reads and remembers the monitor's parameters to eliminate delay or change of video resolution when switching between, or booting computers

Go Online!

For more information, go to iogear.com/lp/3dkvmp

*IOGEAR's DynaSync provides the EDID support for Windows 7

IOGEAR®

Convergence through Connectivity

```
    CPData receivedData;
}

- (IBAction)search:(id)sender
{
}
```

Back in `MainMenu.xib`, connect the `AppController`'s `textField` outlet with the new text field. Then Ctrl-drag from the `AppController object` to the image view and connect it to the `imageView` outlet. Do the same with the table view, connecting it to the `tableView` outlet (make sure you really connect the table view and not the scroll view that surrounds it). Then Ctrl-drag from the text field to the `AppController` object and connect it to the `search` action. Finally Ctrl-drag from the table view to the `AppController` object twice to connect its `dataSource` and `delegate` outlets. The interface is done.

The next thing we need is a data model to represent the search results.

The Data Model

The term "data model" might be a bit exaggerated in this case: We will only create one single class that will represent a single image of a Google Image search result set. To do that, we'll create a file called `GoogleImage.j` in the `image_search` folder. You find the contents of that file in Listing 3. You'll notice that Objective-J doesn't require a class to have an extra header file that makes everything a bit slimmer and easier. Now we just need to connect it all and execute the actual search.

Listing 3: GoogleImage.j

```
@import <Foundation/CPObject.j>

@implementation GoogleImage : CPObject
{
    CPString title @accessors;
    CPString unescapedUrl @accessors;
    CPString tbUrl @accessors;
    int width @accessors;
    int height @accessors;
}

- (id)init
{
    self = [super init];

    if (self)
    {
        title = @"";
        unescapedUrl = @"";
        width = 0;
        height = 0;
    }

    return self;
}

/*!
    Initializes it with the data from a JSON Object
*/
- (id)initFromJSONObject:(id)aJSONObject
{
    self = [self init];
```


Direct Mail

E3 SOFTWARE

jumpstart your small business

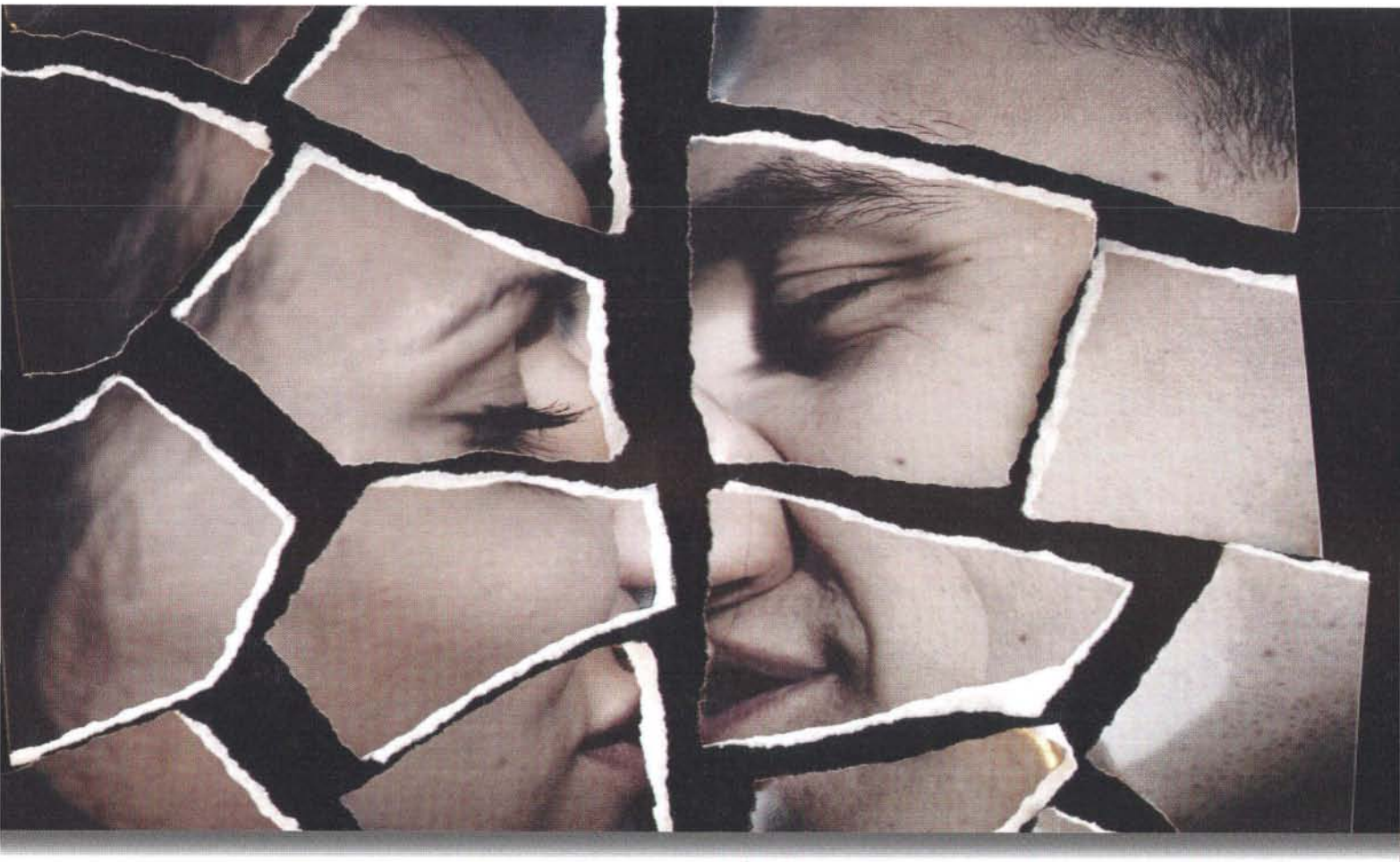
with powerful email marketing

Download for **FREE** Today at www.directmailmac.com

Designed and built only for Mac

- Seamlessly integrates with the Mac applications you already use
- Create and send email campaigns with ease
- Real-time tracking to monitor effectiveness
- **No monthly fees**

SHATTERED MEMORIES?



WE CAN HELP

Recover Your Lost Files Now!

We Provide Data Recovery Software And Services For All Types Of Media

LC TECHNOLOGY INTERNATIONAL

www.LC-Tech.com/mt

A GLOBAL LEADER IN DATA RECOVERY

Mac Universal

```
    if (self)
    {
        // the html entities have to be unescaped
        var e = document.createElement('div');
        e.innerHTML = aJSONObject.titleNoFormatting;
        title = e.childNodes[0].nodeValue;
        unescapedUrl = aJSONObject.unescapedUrl;
        width = aJSONObject.width;
        height = aJSONObject.height;
    }

    return self;
}

- (CPString)size
{
    return width + "x" + height;
}

/*!
    Returns an array of images built from an array of JSON
objects
*/
+ (CPArray)imagesFromJSONObjects:(id)someJSONObjects
{
    var images = [[CPArray alloc] init];

    if (someJSONObjects)
    {
        for (var i=0; i < someJSONObjects.length; i++)
      {
            var image = [[GoogleImage alloc]

initFromJSONObject:someJSONObjects[i]];
            [images addObject:image];
        };
    }

    return images;
}

@end
```

Bringing It All Back Home

Similarly to Objective-C, you have to import source files in Objective-J. This is done with the `@import` directive. To use our `GoogleImage` class in `AppController`, we have to add the line `@import "GoogleImage.j"` to the very top of `AppController.j`.

The `sayHi:` method that we used in the original example can be deleted. The empty stub of the `search:` method has to be replaced with the `search:` method as shown in Listing 4.

Listing 4: AppController: The search method

```
- (IBAction)search:(id)sender
{
    var term = [textField stringValue];
    if (term && [term length] > 0)
    {
        var request = [CPURLRequest
requestWithURL:'http://ajax.googleapis.com/ajax/services/sea
rch/images?v=1.0&rsz=large&imgtype=photo&q=' + term];
        [request setHTTPMethod:@"GET"];

        receivedData = nil;
        [CPURLConnection connectionWithRequest:request
delegate:self];
    }
    else
    {
        alert("Please enter a search term!");
    }
}
```

It reads the search term from the text field, makes sure that it isn't empty and then starts a `CPURLConnection` which queries the Google Image Search API and sets its `delegate` to `self:`, just like you know it from Cocoa's `NSURLConnection`. The next step consists of adding the `CPURLConnection` delegate methods to `AppController.j` as shown in Listing 5.

Listing 5: AppController: CPURLConnection Delegate Methods

```
- (void)connection:(CPURLConnection)connection
        didReceiveData:(CPString)data
{
    if (!receivedData)
        receivedData = data;
    else
        receivedData += data;
}

- (void)connection:(CPURLConnection)connection
    didFailWithError:(CPString)error
{
    alert("Connection did fail with error : " + error) ;
    receivedData = nil;
}

-
(void)connectionDidFinishLoading:(CPURLConnection)aConnectio
n
{
    var res = nil;

    try
    {
        res =
CPJSObjectCreateWithJSON(receivedData).responseData.results;
    }
    catch(err)
    {
        alert("Error while parsing search results: " + err);
    }

    if (res)
    {
        images = [GoogleImage imagesFromJSONObjects:res];
        if (images)
            [tableView reloadData];
      else
            alert("Nothing found.");
    }
}
```

The 3 methods save the received data in the `receivedData` ivar, display a message when an error occurred, and process the received data when loading has finished. The processing is simply done by turning the string we received from the Google API into a JavaScript data structure using the `CPJSObjectCreateWithJSON` function. From the `responseData` object of that data structure we then extract the `results` array. If that was successful, we pass that JavaScript array to the `imagesFromJSONObjects:` class

The People.
The Content.
The *Experience*.

November 2-4, 2011
Sheraton Universal • Los Angeles, CA

http://www.mactech.com/conference/

method of the `GoogleImage` class, thus creating an array of `GoogleImage` objects. That array is then saved in the `images` ivar. Finally, the table view is told to reload its data. In order for that to work, the table view expects two data source methods. Listing 6 contains these methods and one additional method—`tableViewSelectionDidChange:`—that is being called when a different row is selected in the table view.

Listing 6: AppController: CPTableView Data Source and Delegate Methods

```
- (CPInteger)numberOfRowsInTableView:(CPTableView)aTableView
{
    return images ? images.length : 0;
}

- (id)tableView:(CPTableView)aTableView
    objectValueForTableColumn:(CPTableColumn)aTableColumn
                          row:(CPInteger)aRowIndex
{
    var i = [aTableColumn identifier];
    return [images[aRowIndex]
performSelector:CPSelectorFromString(i)];
}

- (void)tableViewSelectionDidChange:(CPNotification)note
{
    var image = [images objectAtIndex:[tableView
selectedRow]];
    var u = [image unescapedUrl];
    [imageView setImage:[[CPImage alloc]
initWithContentsOfFile:u]];
}
```

The first method returns the number of rows in the table view that corresponds to the number of elements in the `images` array. The second method uses the `CPSelectorFromString` function to create a selector from the column identifier that was passed in to identify which column data is being requested for (either `title` or `size`) and calls that selector on the corresponding `GoogleImage` object and finally returns the result. So this method either returns the title or the size of the image for the requested row. The third method finds the `GoogleImage` object of the selected row and loads the corresponding image in the image view.

That's it: The application is done! When you now reload index.html in your browser and search for "cappuccino.org" everything should look kind of like Figure 6.

Figure 6. The finished image search application

Conclusion

Cappuccino is a very powerful framework that lets you develop beautiful applications. You don't have to learn a slew of new languages or technologies but can reuse and expand on your valuable knowledge of Objective-C and Cocoa. So what's stopping you from porting your Cocoa application to Cappuccino now?

Lost Data?

DRIVESAVERS® DATA RECOVERY

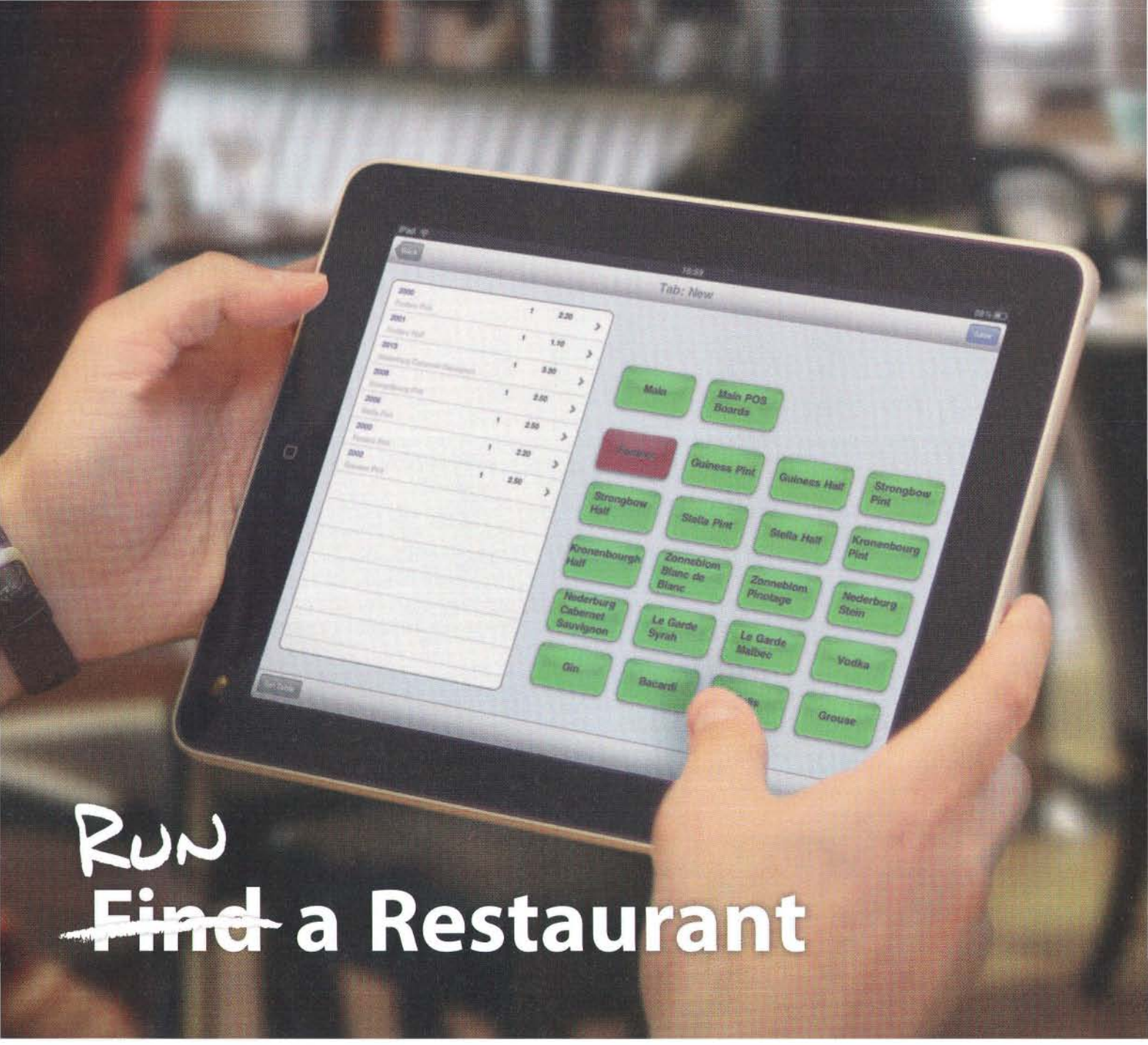
We can save it!

Fast • Reliable • Certified
Secure • Data Recovery

800.440.1904
www.drivesavers.com
©2010 DriveSavers Data Recovery Inc.

About The Author

Johannes Fahrenkrug is a freelance software developer, co-author of the book Objective-C Fundamentals, frequent speaker at conferences and passionate JavaScript and Cocoa developer. You can reach him at springenwerk.com or on Twitter @jfahrenkrug.



RUN ~~Find~~ a Restaurant

Ordering | Kitchen Printing | Reporting

Tabs opened from graphical table view. Fast, accurate orders, including special requirements. Wireless printing to kitchen and for final bill. Improve your customer service with Restaurant by HansaWorld, for iPad.

Books by HansaWorld

Enterprise by HansaWorld

Integrated CRM | Real-time Mobile | Real-time Consolidation | Integrated Webshop | Built-in Document Management | Powerful Analysis

HansaWorld USA, Inc. 1501 Broadway #12067 & 12068, 12th Floor, New York, NY 10036-5601, USA, Phone: +1 646-459-0603, Email: usa@hansaworld.com

by Boisy G. Pitre

No Pane, No Gain

Building a Preference Pane plug-in to integrate into System Preferences

Introduction

Have you ever been curious about extending the System Preferences application on your Mac? System Preferences is the central location for changing and fine-tuning a wide range of options, from display settings to networking to mouse settings, and everything in between. You've probably noticed that from time to time, the installation of a new application causes a new icon to appear in the System Preferences window. This is possible because Apple has architected System Preferences to accept 3rd-party preference pane plug-ins, and in this month's *Developer to Developer*, we'll go through the steps to construct our very own preference pane plug-in. As always, a ready-to-run project is available on MacTech's FTP site.

Why Preference Pane Plug-ins?

When developing a GUI-based application, we typically setup a preferences window that allows the end-user to control and customize settings. After the application terminates, the preferences are stored away in a file so that subsequent launches of that same application can consult and adjust to those settings. Usually, there is no need to adjust those settings when the application isn't running.

A preference pane plug-in, on the other hand, allows users to change settings outside of the scope of a single running application. Hence, the System Preferences application focuses on settings that are global in nature and affect the running of the operating system. When determining whether your application needs a preference pane plug-in, ask yourself the following questions:

Does your application perform functions that affect the system as a whole?

Is your application a non-gui one (i.e. launch daemon) whose settings need to be changed by the user?

If the answer to either of these questions is yes, then in all likelihood, a preference pane plug-in is the correct venue for your needs. In fact, a common use of a preference pane plug-in is to provide some visual form of interaction for a daemon or other low-maintenance, non-GUI process running on the system.

Before we launch full-scale into the process of creating a preference pane plug-in, let's take some time to learn about the System Preferences application itself.

System Preferences Then And Now

On all Macs, the System Preferences application resides in the `/Applications` folder; it can be launched from there, or quite conveniently, from the Apple menu at the top left corner of your main screen.

Up to Leopard (10.5), System Preferences was a 32-bit application, and all preference pane plug-ins were also 32-bit. Starting with Snow Leopard (10.6), it became a 64-bit application that can accept both 32-bit and 64-bit plug-ins. When a 32-bit preference pane plug-in is encountered in 64-bit mode, System Preferences will relaunch itself in 32-bit mode to adapt to the plug-in.

At this point, all preference pane plug-in developers should be building both 32-bit and 64-bit versions of their plug-in. Building it in this way will ensure that the plug-in runs smoothly on both environments.

Where Plug-Ins Live

Preference pane plug-ins are special bundle files which live on your Mac's file system. They have an extension of `.prefPane` and contain a number of files which we will go over shortly. All Apple-supplied preference pane plug-ins are in the `/System/Library/PreferencePanes` folder. If you're curious, feel free to examine this folder, but be careful not to modify anything, since these plug-ins are part of the standard system install.

For 3rd-party preference pane plug-ins, Apple has dictated that they reside in either one of two folders:

```
/Library/PreferencePanes
~/Library/PreferencePanes
```

The first path is a global one; preference pane plug-ins that reside in this folder will be seen by all users on your system. In general, this is the appropriate place for preference pane plug-ins to reside. However, as option 2 above indicates, they can also be installed inside of a user's home folder. Here, the preference pane plug-in will only be seen by that user when he launches System Preferences.

Installing A Preference Pane Plug-in

The installation of a preference pane plug-in is quite simple. Just double-clicking on the file in Finder is enough to launch

System Preferences. At that time, the preference pane plug-in bundle will be examined and System Preferences will ask if you want to install the plug-in for the current user or for all users. Depending upon your answer, it will either copy the bundle to your home directory location, or the root location where all users can view the plug-in. If you choose for all users to utilize the plug-in, you will need to type in the administrator password since the copy operation requires admin privileges.

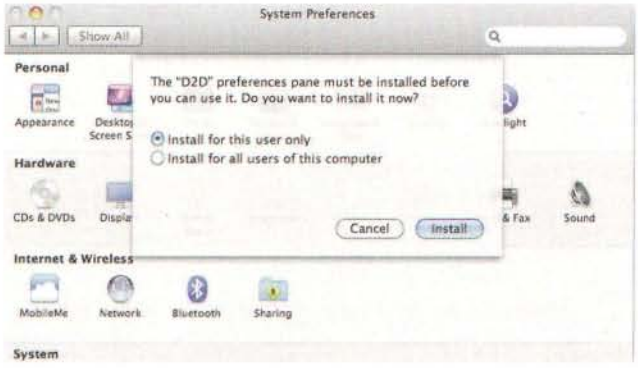


Figure 1. System Preferences Installation Option

Peeking Into The Code

Now that we have explored System Preferences a bit, let's take a look at the details of implementing a preference pane plug-in. If you haven't already done so, download the project source file `D2DPrefPane.zip` from the MacTech FTP site and decompress it on your hard drive, then open up the `D2DPrefPane.xcodeproj` file in Xcode.

The project consists of a number of source files along with resource files such as icons and images. The following table lists each of them and explain their purpose:

Table 1. Project Files and their Purpose:

File name	Purpose
D2DPref.h / D2DPref.m	This is the home of the D2DPref Objective-C class. It is subclassed from the NSPreferencePane class and holds the code for our plug-in.
D2DIcon.png	This is the icon that will appear in the System Preferences application.
MacTech.png	This is a graphic that will appear in a button on our preference pane plug-in.
Info.plist	The property list file that holds important information about the structure of our plug-in.
D2DPref.xib	The Interface Builder file which contains the visual layout of our preference pane plug-in.

Now let's examine the central source file to this project, `D2DPref.m`. When developing a preference pane plug-in, you must have an Objective-C class that is subclassed from `NSPreferencePane`, and your subclass must implement the `mainViewDidLoad` method. This method is the entry point into your custom preference pane plug-in, and can contain any initialization code that you need to perform. Since our preference pane is simple, we currently do nothing here.

A custom method that we have defined in this file is `urlButtonAction:` which is an action method for a button that exists in the preference pane. When the button is clicked, a browser will be spawned to take us to the website of our favorite magazine.

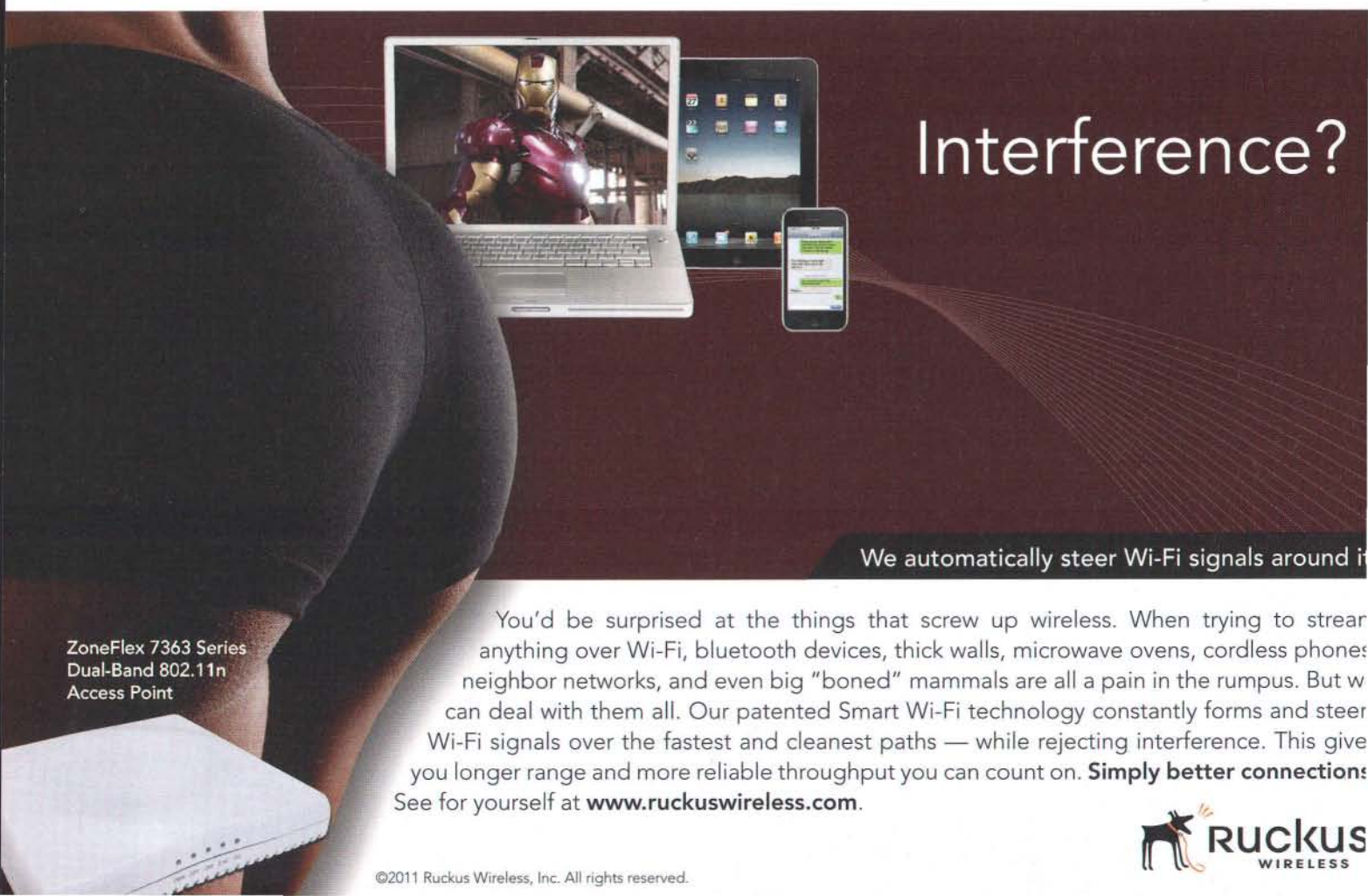
Looking Into The Nib FIle

Now open the `D2DPref.xib` file in Interface Builder. The File's Owner for this NIB file is the `D2DPref.m` file that we just examined. Since `D2DPref` inherits from NSPreferencePane, we must hook up the _window outlet to the `PrefPane` window object within the NIB file. That is already done, but you can confirm by right-clicking on the File's Owner icon to see that the connection is made.

Double-click on the `PrefPane` object in the NIB file document and you will see that the object is an NSWindow with a single button in the middle containing the graphic of our favorite magazine. If you examine the action outlet of this button, you will see that it points back to the `urlButtonAction:` method in our `D2DPref` class (which is the owner of the NIB file). So the expectation is that if we click on the button, it should invoke the action method that launches a browser and takes us to the previously indicated URL.

Figure 2. The preference pane window in Interface Builder

There is something important to note here: the dimensions of the NSWindow are specific, particularly the width. Starting with Leopard, all preference panes are to be 668 pixels wide, somewhat larger than previous versions of System Preferences on Tiger and prior operating systems. The height can vary, but your NSWindow object should maintain a width of 668 pixels, as we do here.



Build And Go!

Now here's the moment we've all been waiting for. Switch back to Xcode and build the project. Once it is built, locate the Products folder in the Xcode Groups & Files pane. There you should see the D2D.prefPane bundle. Double-click and it will launch System Preferences with the dialog box asking how you wish to install the plug-in. For now, simply install it for yourself only. System Preferences then copies the bundle to your home directory's `Library/PreferencePane` folder and launches the plug-in. You should now see the MacTech button sitting solitary in the middle of the pane. Go ahead and click it; it should take you to the MacTech website.

Summary

This month, we've examined the operation of the System Preferences application along with its plug-in architecture. From there we successfully built a preference pane plug-in from the provided Xcode project and installed it on our system.

One issue that hasn't been addressed thus far is that of security. Depending upon the application, there may be times when it is necessary to access or even modify files that belong to root. Next month, we'll extend our preference pane plug-in with authentication services and the ability to run external commands as root. We'll also throw in some other nifty tid-bits about preference pane plug-ins and discuss how to use the Xcode debugger to debug your plug-in. Until then, I suggest reading the documents in the bibliography below to expand your knowledge of preference panes. Until next month, have fun!

Bibliography and References

Apple, *Preference Pane Programming Guide*,
http://developer.apple.com/library/mac/documentation/UserExperience/Conceptual/PreferencePanes/PreferencePanes.pdf

Cocoa Dev Central, *Create a Preference Pane*,
http://cocoadevcentral.com/articles/000030.php

About The Author

Boisy G. Pitre lives in Southwest Louisiana and is the lead developer at Tee-Boy where he also consults on Mac and iOS projects with a variety of clients. He holds a Master of Science in Computer Science from the University of Louisiana at Lafayette. Besides Mac programming, his hobbies and interests include retro-computing, ham radio, vending machine and arcade game restoration, and playing Cajun music. You can reach him at boisy@tee-boy.com.

1

web and mobile recorder works on the Mac

As an IT professional, you need reliable website performance monitoring solutions that work seamlessly with Macs. DéjàClick™ is the only inside-the-browser transaction monitor that runs and tests site transactions like a real user—and it's the only one that runs natively on the Mac. AlertSite's industry-leading TrueUser™ technology identifies slowdowns and other frustrating site problems—before your customers do. Take a fresh look at your website with a free trial at AlertSite.com/mactech. We'll give you a $15 Starbucks gift card for giving it a try.*

Your customers count on you. Count on AlertSite.

AlertSite®
A SMARTBEAR BUSINESS

AlertSite.com/mactech 877.302.5378

© 2011 SmartBear Software. All rights reserved. *Must pre-qualify to receive offer

Enterprise Desktop ALLIANCE

Scripting Automations in Windows for Mac OS X Administrators

by Charles Edge

Introduction

To minimize the Total Cost of Ownership (TCO) of large numbers of systems requires some form of customization. Such customization may come in the form of scripting simple tasks, automating complicated procedures and usually includes building bridges between disparate systems using the APIs available for those systems.

The growth of Apple in large environments means that there are more and more enterprises looking to adopt a platform that allows the enterprise to not only provide services similar to how they are provided for other platforms, but also allows the enterprise to centrally manage the client computers. The Enterprise Desktop Alliance (EDA) has developed a strategy for leveraging existing Windows Server administrators and infrastructure in order to provide command, control and connectivity services to Mac OS X clients. The EDA includes Absolute Manage, Centrify, GroupLogic, IBM and WebHelpDesk, all able to run on Microsoft Windows Server 2008 R2. Each of these vendors fills a very specific void for managing Mac OS X:

Absolute Manage: Change, configuration and patch management

Centrify: Extending group policy objects to enable Active Directory-based management and a directory services plugin to ease the transition to Active Directory

GroupLogic: Native Mac OS X file services hosted on Windows Servers

IBM: Web and groupware services as well as highly available Windows Server hardware

WebHelpDesk: Trouble ticketing and inventory management

This ecosystem provides systems that work very well together, or independently, maximizing the efficiency of staff and giving administrators repeatable, highly available, well documented and vendor supported infrastructures. The EDA has run 3 previous articles in MacTech Magazine, outlining options for replacing the Xserve with Windows Servers now that Apple no longer provides rack-dense solutions. This allows administrators to run their most critical and important services that enterprises need, even while the Xserve is deprecated. These articles included:

January 2011: Centralized Mac Home Directories on Windows Servers

February 2011: Imaging and Patch Management using Windows Servers for Mac OS X Clients

March 2011: Implementing File and Print Services for Mac OS X using Windows Servers

April 2011: Large Scale Mac OS X Client Management Using Windows Servers

In this, the 5th installment of the EDA series on moving from Mac OS X Servers to Windows Servers, we will look further at ExtremeZ-IP, Absolute Manage, Centrify and WebHelpDesk. Our focus this month will be on extending what these products can do in order to tailor them to your unique environment. We are moving beyond the graphical options that the vendors provide and into more of a command line and scripting environment.

As we have been showing throughout this series of articles, the move from Mac OS X Server to Windows Server can be less cumbersome than many previously thought. The platform is considerably more scalable, with virtualization end-to-end and true high availability options. And in many environments, Active Directory has been integrated for years and so administrators are already well versed in Windows Server administration basics. The impact of replacing systems on existing middleware components though, can be amongst the most impactful. In this article we will take a look at some common automations that have been built for Mac OS X environments and look at ways to port them over into Windows server environments so that organizations can have the level of rack density, failover and scalability that they require while lessening the amount of scripting that is required.

Middleware

Got Macs?
Got Windows Servers?
Get them working together.

Administrator's Windows Environment

Absolute Software	Centrify	ExtremeZ-IP	Web Help Desk
Client Configuration	Active Directory Integration Group Policy for Mac	File Server & Print Server	Help Desk & Knowledge Base

User's Mac Environment

It actually IS easy to deploy, integrate and manage Macs in a Windows environment. Visit us to learn how:

www.enterprisedesktopalliance.com

©2010 Enterprise Desktop Alliance

Enterprise Desktop ALLIANCE

No discussion of building network infrastructures to scale can be complete without discussing middleware. This is because most solutions fit within an organization to solve a specific pain point. Middleware allow administrators to interconnect and customize various solutions in such a way that the whole is much greater than each part. The customizations save time, increase productivity and allow each component to be built in such a way that it can scale in ways it otherwise could not.

Every environment is different. As such, every environment has different needs. Most organizations will need multiple solutions and these solutions have different features that each organization will customize in ways that make sense for them. But often times, the solutions will not communicate with other solutions meaning that administrators have to do entry into different solutions manually.

For example, let's look at a typical school environment. Each student, when enrolled, is entered into a student information system. The student is then entered into a directory service, such as Active Directory or Open Directory. Once entered, the student has a profile created for them and has a set of policies applied to their account. The student is also placed into a group or multiple groups, based on grade, school they attend, etc. The student may have an email or groupware account created for them as well as be entered into a course management system. Each year that the student attends the school their grade will change and according to how the systems are setup they may move between different groups (e.g. elementary school, then junior high, then 10th grade, then 11th, then 12th and ultimately archived).

If a school has a lot of students then this can result in a massive amount of work for administrators. However, provided that a consistent user experience is needed, a few scripts can result in drastic reductions in the amount of time spent managing students. This can include automatically creating the user in the directory service, assigning group membership, allocating a computer (as is common in a 1 to 1 deployment) and even archiving the student when they have graduated or transferred. This type of automation allows a small number of administrators to manage even large districts.

Exposing the API

Middleware makes administrators more efficient. But the capacity to script the middleware is dependent on the APIs made available from software vendors. An API, or Application Programming Interface, is comprised of the tools and other resources a vendor makes available to extend its software. The vendors comprising the Enterprise Desktop Alliance each have a number of automations that commonly revolve around their solutions to aide when managing the lifecycle of systems.

In order to integrate the Web Help Desk with other applications, HTTP interfaces are provided to log clients into the application (skipping the login page) and to create tickets. The Web Help Desk links can then be accessed from other solutions, automatically generating tickets. Administrators can also edit the Web Help Desk database directly, although when it is possible to augment a solution using an API to accomplish a given task, it is always preferable to do so over editing the solutions back end directly.

The Centrify DirectControl API allows administrators to control zones and NIS maps. Zone control includes programmatically creating, editing and deleting zones within Centrify as well as adding users and groups to these zones. This allows administrators to bolt Centrify into other solutions, such as Student Information Systems, Identity Management Systems and even other middleware components already in use and development.

The Centrify DirectControl API provides standard Windows COM objects that convert Active Directory application objects into Centrify-enabled UNIX user, group, computer, and zone objects. These are packaged in the DLL (Dynamic Link Library) that is distributed with the DirectControl SDK. Documentation for the API can be found at http://www.cerberis.com/images/produits/techFiles/Centrify-DirectControl-4-Programmer-Guide.pdf.

ExtremeZ-IP has a mature and full-featured API whereby almost complete control of the application has been exposed by the API. Using the API, it is possible to create, edit and delete print queues and file shares within ExtremeZ-IP. There are a number of different ways to interact with ExtremeZ-IP programmatically, but none is easier than controlling the options exposed in the EZIPUTIL.exe command line options. But given that for every scripter there is a different way of doing things, ExteremeZ-IP has also provided access via C++, WMI and even a web services API. Later in this article we will look at using the command line in a scripted workflow and provide links for accessing even more information as needs progress.

Absolute Manage can also be accessed using an API of sorts. Primarily, interacting with Absolute Manage will take form of reading data from a MySQL database, allowing administrators to query information indirectly from Absolute and leverage the output of those queries in other scripts and data structures.

Controlling Client Computers

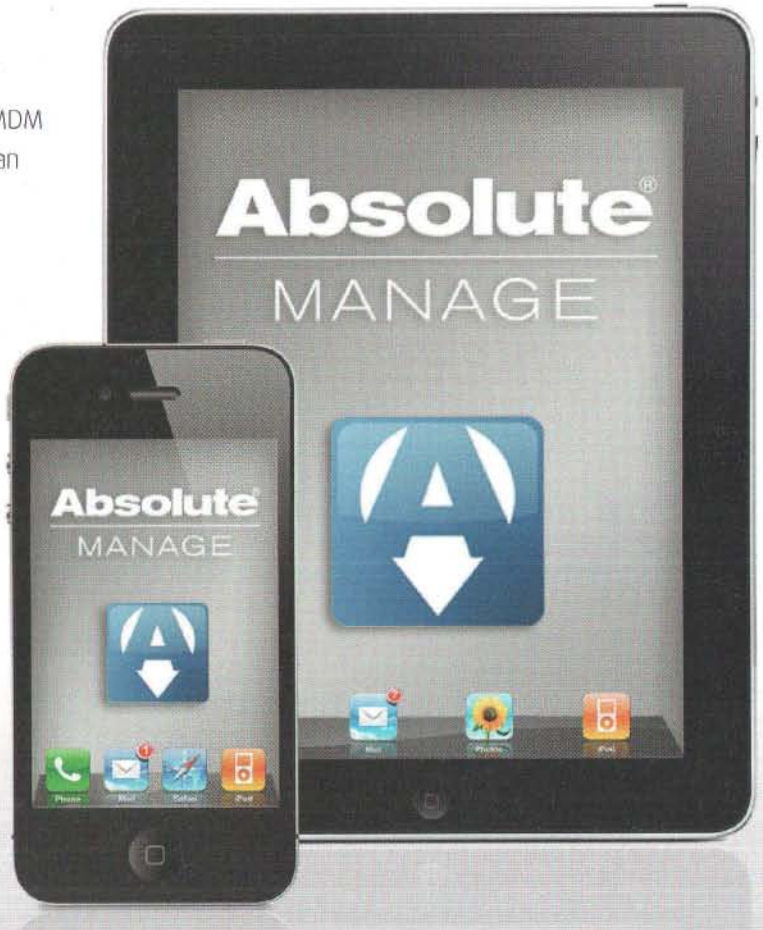
Absolute Manage has registered the lanrevagent:// handler when the Admin installer is installed. The syntax for invoking lanrevagent is similar to calling up a web page or an AFP mount, except here you define the command, followed by what exactly to run that command against. The command to remotely control a host is remotecontrolagent. The easiest way to indicate an agent to be controlled is to define a computername that the agent can control, done by following the remotecontrolagent command with `?computername=<the actual computer name to be used>`. When defining `computername`, you will need to replace any special characters with their URL encoded representation (e.g. – for a space that would be a %20). This is really actually very straightforward. To control a computer named Charles Edge MacBook Air, you would use the following URL from a browser:

MANAGING YOUR MOBILE APPLE DEVICES JUST GOT EASIER

Absolute Manage Device Management for iOS4

Our technology is designed to work within a Windows or Mac environment so you can use whatever you already have in place.

And your IT administrators will love you since Absolute Manage MDM can be driven using a Mac or a PC. Now everyone on the team can be an Apple iOS4 expert!



Asset Inventory

Gather 60+ hardware and software data points and integrate the data into third party applications (SCCM, Web Help Desk, etc.)

Data & Device Security

Lock a device, clear a password, wipe a device clean, manage and deploy profiles, send messages to end users

Application Management

Track installed applications, collect data from each device, securely host and deploy in-house applications

Try it yourself! Visit our product evaluation page:

www.absolute.com/MacTech

© 2011 Absolute Software Corporation. All rights reserved. Computrace and Absolute are registered trademarks of Absolute Software Corporation.

Absolute®Software

The absolute best way to track, manage & protect your digital world.

```
lanrevagent://remotecontrolagent?computername=Charles%20Edge%
20MacBook%20Air
```

Or simply use the terminal command:

```
open
lanrevagent://remotecontrolagent?computername=Charles%20Edge%
20MacBook%20Air
```

Using the open command allows for variable substitution, which makes this yet another addition to the possibilities surrounding automating administration tasks through help desk software without having to expose passwords to intermediate administrators. Provided that host names are in synchronization between servers, you then have the ability to invoke commands against Absolute Manage using other products as part of your management lifecycle or some form of middleware.

To restart the Absolute Manage service,

```
net stop "LANrev Server"
```

Followed by:

```
net start "LANrev Server"
```

On the server, administrators can also start and stop the Absolute Manage Agent and Absolute Manage Server in Windows using the Windows Services applet or do so programmatically with a registry edit. To restart the Agent or Server (and therefore reload configuration data), the Trigger registry key would need to be set to 1. These are located in HKEY_LOCAL_MACHINE\Software\Pole Position Software\LANrev Agent\Trigger and HKEY_LOCAL_MACHINE\Software\Pole Position Software\LANrev Server\Trigger registry keys for the Agent and Server respectively.

ExtremeZ-IP can also be started and stopped programmatically. To restart ExtremeZ-IP:

```
net stop "ExtremeZ-IP"
```

Followed by:

```
net start "ExtremeZ-IP"
```

But service control is only the beginning of what can be done using the command line when controlling ExtremeZ-IP. Shares, printers and server settings can also be configured.

Creating Shares Programmatically

Windows and Windows Servers have a command line environment similar, albeit far less functional from a scripting perspective to the shell environment in Mac OS X. Scripts, known as batch scripts can then be written to automate basic functionality in Windows Servers. While you can do a lot with PowerShell or WMI, batch scripts are simple and quick to write and execute. For administrators already familiar with Python, Ruby, Perl and other languages more native to Mac OS X, those can all be used in Windows as well. ExtremeZ-IP can be controlling using the EZIPUTIL command. This is similar to how the serveradmin command can be used to set global settings in Mac OS X Server. For example, to start the server services, the SERVER option can be used in conjunction with the /START switch:

```
EZIPUTIL SERVER /START
```

The SERVER option can also be used for a number of other tasks, such as obtaining a list of files in use through ExtremeZ-IP:

```
EZIPUTIL SERVER /FILES
```

Or to see which users are currently logged into the system:

```
EZIPUTIL SERVER /USERS
```

The SERVER option is for controlling global information about the server. The VOLUME option can be used to create, edit and delete shared volumes through ExtremeZ-IP, much as the sharing command can do so in Mac OS X Server. As shown in previous articles, there are a number of settings that can be used for volumes. Each of these is available using the EZIPUTIL command. The following are a few switches for shares that are available from the EZIPUTIL command.

/ADD – Creates a new shared volume

/EDIT – Edits an existing shared volume

/NAME:volumename – Configures the shared volume's name

/PATH:root directory path – Sets the path of the file system that is shared out

/READONLY:TRUE|**FALSE** - Makes the share read-only

/GUESTSALLOWED:**TRUE**|FALSE - Allows guest access to the share

/PASSWORD:password – Enables volume-based passwords

For example, to create a volume for a path of c:\SHARED\ACCOUNTING (the /PATH switch) where guests are not allowed (the /GUESTSALLOWED switch) where Time Machine volumes are allowed (the /IS_TM_VOLUME switch), the following command would create such a share:

```
EZIPUTIL VOLUME /ADD /NAME:ACCOUNTING
/PATH:c:\SHARED\ACCOUNTING /GUESTSALLOWED:FALSE
/IS_TM_VOLUME
```

Scripts leveraging the EZIPUTIL command can then be crafted to perform a number of tasks, such as automatically creating a share for each user as users are created in another system, creating shares for specific groups and even deleting shares at the end of their lifespan as part of a Information Lifecycle Management policy.

The third and final option provided by the EZIPUTIL command is the PRINT option, which surprisingly allows for the management of print queues. PRINT has the same basic switches in /ADD, /EDIT, etc. But the other switches are specific to managing printers. A sampling includes the following:

/PPD – Allows providing a PPD to supply Mac OS X clients with a driver

/PPD_ONLY_FROM_SERVER:TRUE|**FALSE** – Forces clients to only use the PPD provided by the server

WINDOWS – Queues are available as standard Windows printer Queues

/PRINTER:printer

LPR – Queues are available as LPR

/HOST:host -

/QUEUE:queue

In the following example, we will create a printer from the command line, using an existing printer installed in Windows called Finance_LaserJet. This printer will have the same queue name through ExtremeZ-IP that is used to connect Windows users and will require Mac users to download a PPD in order to use it that is stored at c:\Drivers\PPDs\LaserJet_P4041n

```
EZIPUTIL PRINT /ADD /METHOD:WINDOWS
/PRINTER:Finance_LaserJet /PPD:
c:\Drivers\PPDs\LaserJet_P4041n /PPD_ONLY_FROM_SERVER:TRUE
```

To find more information on these, you can use the EZIPUTIL command followed by the HELP verb:

```
EZIPUTIL HELP
```

While administrators can use ExtremeZ-IPs command line tools to create shares, it can be somewhat unwieldy to leverage a share per user for creating home directories, and so administrators often use a share for a group of users and create folders for each user based on, for example, Active Directory group or Organizational Unit membership.

Automated Policy Management

Visual Basic is a language that is often used for automating basic tasks in Microsoft Windows Server environments. Visual Basic is similar in many ways to perl or python and can be as simplistic or complicated as each task that needs scripting. This section of the article isn't a guide on writing Visual Basic scripts, a topic many full books currently explore. Instead, this section of the article is meant to showcase how to do a few basic tasks that Mac OS X administrators have been doing with scripts in Mac OS X environments for years.

Managed preferences allow for policy-based management of Mac OS X clients. In previous articles, we looked at leveraging Centrify to provide such management. Centrify, as mentioned earlier in this article, also allows for a programmatic interface. The interface is available in VBScrpt, J Script, and .NET via the Centrify SDK (CentrifyDC_SDK-release-win64.zip), available at Centrify.com.

Installation of the SDK is straightforward, choosing the default options at each screen. Once the SDK has been installed, a number of methods to work with Centrify will be available in Visual Basic. For example, the following can be used to output a list of users that are in a given zone. This is done using the cims.getzone method from the Centrify SDK. We will also use a for loop, looping through all of the users in the zone and ultimately echo the output:

```
set zone = cims.getzone("krypted.com/program data/centrify/zones/default")
for each user in users
wscript.echo user.name, user.Uid
next
```

The output would provide short names followed by unique IDs for users, as follows:

```
ledge 10088
cedge 10089
eedge 10092
```

This is a very simplistic script, with much more being made available by Centrify as part of the SDK. A script like this could be used to quickly query for all of the users that have a given set of settings and using other methods it would be possible to add users to zones or augment their directory data with information specific to Centrify.

Creating Folders Within Shares

As we've shown in previous articles in this series, Active Directory, using Centrify, can provide the necessary back-end infrastructure for centralized home directories without any Mac OS X Servers. One of the more common tasks that needs to be scripted during directory services migrations and during the setup of new users is to create the folders that these new and migrated users will use when logging into accounts.

The User Home Creator script, seen below, showcases how to work with the FileSystemObject object and its CreateObject method, which can be used to (as you can probably guess) create an object on a filesystem. In VBScript, creating an object is done in a single line (it is common Visual Basic practice to prefix objects with obj and strings with str). Here, we can create a FileSystemObject called objFSO:

```
Set objFSO = CreateObject("Scripting.FileSystemObject")
```

We would also create a network object (which we call objNetwork). We then create an array of a collection of items queried from a given Organizational Unit (here, specified as Users) in Active Directory and then loop through those items and create a directory if one does not yet exist. Finally, we call on Wscript.Shell to run a shell command, using the SetACL.exe executable to change the permissions on the folders that we create.

User Home Creator

```
'Create our FileSystemObject and our Wscript.Network object

Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objNetwork = CreateObject("Wscript.Network")

'Create an array of users in the OU

Set colItems = GetObject _
    ("LDAP://ou=Users,dc=318,dc=com")
colItems.Filter = Array("User")
```

```
'Loop through the array, read the user into strUser and the
path to the homes into strDest, creating the folder and
generating permissions

For Each objItem in colItems
    strUser = objItem.sAMAccountName
    strDest = "\\afp03.318.com\homes\" & strUser
    Set objFSO = CREATEOBJECT("Scripting.FileSystemObject")
    IF Not objFSO.FolderExists(strDest) THEN
        Set objFolder = objFSO.CreateFolder(strDest)
        strDest = "\\afp03.318.com\homes\" & strUser
        Set objShell = CreateObject("Wscript.Shell")
        objShell.Run ("\\\\afp03.318.com\netlogon\\SetACL.exe
-on """ & strDest & """ -ot file -actn ace " & "-ace
""n:AD\" & strUser & ";p:full""")
    ELSE
    END IF
Next
```

This script shows off the access to objects, IF/THEN and arrays. There are likely more elegant ways to perform these actions, but overall this shows basic functionality of the Visual Basic scripting environment in a way that is beneficial when creating network directories to be used for mobile homes.

Bridging Applications

Absolute Manage allows other systems to view its data in the form of an exported MySQL database. Absolute Manage stores its data in SQLite, rather than MySQL, but administrators should not interact directly with the SQLite database (it is not made available for network access anyway). Instead, data can be made available via MySQL for other applications using an ODBC export for MySQL. That's essentially a one-way street; inventory data goes out but it doesn't come in.

Web Help Desk has a built-in discovery connection that can use information from the ODBC exported MySQL data to create asset entries to keep administrators from having to perform double entry. Bringing Absolute Manage data from MySQL into Web Help Desk to create new asset entries allows administrators to view inventory details about client's in Absolute Manage admin to initiate a remote control sessions using a link from Absolute Manage in the Web Help Desk console.

To get started, first install and configure a MySQL server that will be connected to by Web Help Desk. Then create a database in MySQL to host the inventory data. Once there is a database on the server that can accept the ODBC export, install

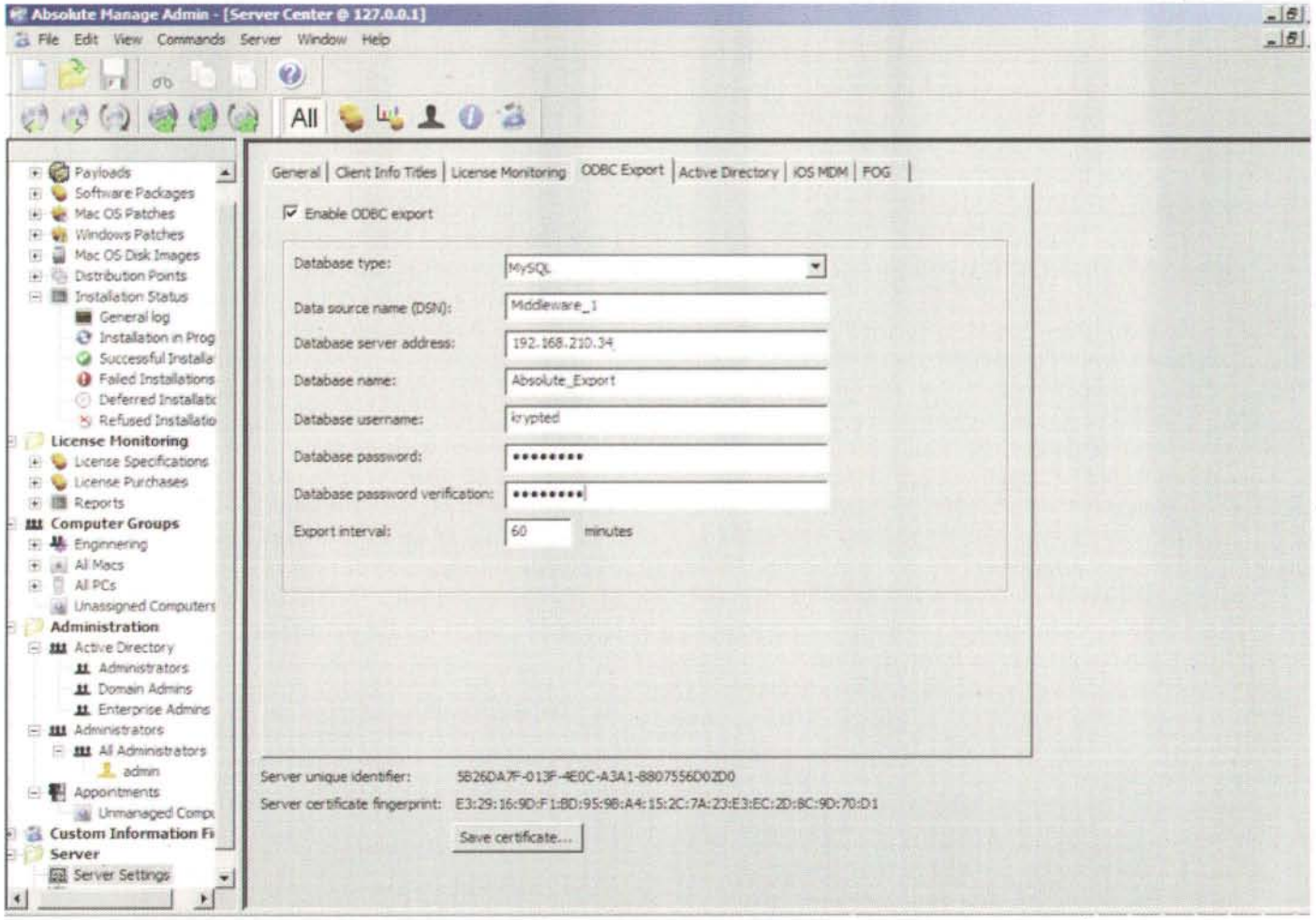


Figure 1 – Creating an ODBC export.

For 20 years now, Group Logic solutions have been helping Macs and PCs get along. Over 100,000 Mac users rely on our ExtremeZ-IP solution to seamlessly access Windows-based digital assets without compromising performance, usability, or reliability. To learn more how we can help you, please visit us on the Web at **www.grouplogic.com** or call **800.476.8781 / +1.703.528.1555.**

FREE TRIAL
www.grouplogic.com/eztrial

GroupLogic.
Access. Share. Extend.

© 2010 Group Logic. All Rights Reserved.

Group Logic is a Proud Member of

Absolute Software · Centrify · ExtremeZ-IP · Web Help Desk · Enterprise Desktop ALLIANCE

the MySQL ODBC driver on the Absolute Manage server. To do so

Open the Absolute Manage Admin application on the Absolute Manage server.

Choose Server Center from the Window menu.

Click on Server Settings in the side bar.

Click on the ODBC Export tab.

Check the box for Enable ODBC export tab.

Provide details for the ODBC connection:

Data source name (DSN):

Database server address: The address of the server in question.

Database name: The name of the database on the server.

Database username: The name of the user configured to authenticate to the server.

Database password: The password of the user that has been configured.

Database password verification: The password of the user a second time for verification.

Export interval: The number of minutes between database exports.

Once the required information has been provided, open MySQL and verify that the databases have exported as intended. Provided that they have, it is then time to integrate a 3rd party with the database. There isn't currently any public documentation of the MySQL database structures. But generally the foreign key columns are consistently named <base_table_name>_record_id, where <base_table_name> is the name of the table that the foreign key references. Within this table, the foreign keys generally reference the "id" column. For example, the hardware_info.agent_info_record_id is a foreign key referencing agent_info.id.

There are a few exceptions to this rule. The base table for all computer-related tables is the agent_info table, all other tables relate to this table using the 'agent_info_record_id' column as a foreign key referencing agent_info.id. Some columns are enumeration values, in which case the value relates to a corresponding enumeration table. Enumeration table names have a prefix of "enum_" and a postfix that loosely relates to the column name in the table containing the enumeration values. For example, ata_info.DeviceType -> enum_ATADeviceType and agent_info.AgentPlatform -> enum_AgentPlatform. The names of the custom fields can be associated with the custom field values as agent_custom_fields.FieldID=custom_field_names.id.

The other groups of records that have independent relations include:

iPhone records:

iphone_info is the base table

iphone_installed_software_info relates to iphone_info.id using the iphone_info_record_id column

Software distribution:

sd_packages contains package descriptions

sd_payloads contains package payload descriptions

sd_groups contains computer group descriptions

sd_staging_server contains staging server descriptions

sd_diskimages contains disk image descriptions (disk images for software distribution purposes)

sd_metapackages_packages lists the packages contained in the metapackages

sd_groups_packages associates sd_groups with sd_packages

sd_groups_agents associates sd_groups with agent_info records (for plain computer groups)

sd_groups_staging_servers associates sd_groups with sd_staging_server records

sd_installation_status contains installation status info and references sd_packages via sd_installation_status.sd_package_record_id=sd_package.id

License monitoring - tables having an 'lc_' prefix, e.g. lc_license_specs contains the license specifications, lc_groups_specs relates sd_groups to lc_license_specs records, etc.

Administrator setup and assignment - tables "admins" and all tables with an "admin_" prefix where "admins" contains the administrator accounts:

admin_apppointments contains the definitions of the appointment groups

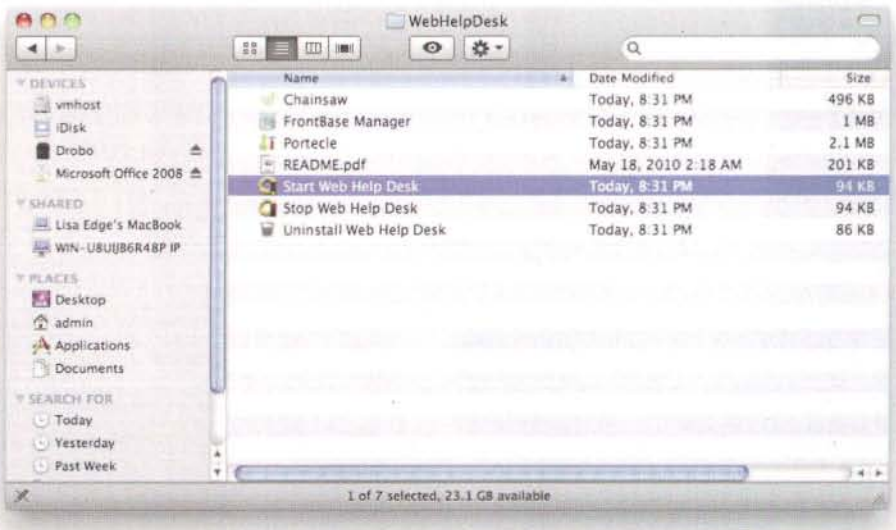


Figure 2 – Web Help Desk install.

admin_appointments_admins associates admins with appointment groups

admin_appointments_agents associates computers (agent_info records) with appointment groups (for plain appointment groups only), where this relation is done through the agent serial number instead of agent_info.id, i.e. admin_appointments_agents.AgentSerial=agent_info.AgentSerial

admin_groups contains administrator group definitions

admin_groups_admins associates admins with admin groups

In this case, we will look at using our data to populate Web Help Desk, which can pull client inventory data from the MySQL inventory database. To do so, configure an asset discovery connection with the correct access settings to connect to Web Help Desk. First, install Web Help Desk, following the default settings during the installation process. Then, open the Web Help Desk directory within /Applications and open the Start Web Help Desk application.

Next, from within Web Help Desk:

Click on the Setup Icon.

Click on Assets (assets is only available in the Full version, so for environments running the Lite version, a switch will need to be made to the Full version).

Click on Discovery Connections.

Click on New.

Provide connectivity information for MySQL:

Connection name: A friendly name for the connection.

Discovery tool: Set to Absolute Manage (LANrev).

LANrev MySQL Database host: The address of the MySQL server.

Port: The port that MySQL runs on.

Database name: The name of the database provided in the earlier part of this article.

Username: A username with appropriate access to the database that was created earlier in this article.

Password: The password provided in the Username field.

Include Virtual Machines: Choose whether virtual machines should be included in the import.

Auto-sync Schedule: Set a schedule to perform imports, allowing administrators to import database content automatically.

Ignore Black Discovered Values: Allows for manual editing once data has been imported into Web Help Desk.

Client Relationships: Determines how users and computers are associated.

When Assets Are Removed: If set to Delete Asset then as assets are removed from the Absolute database so to shall they be removed from Web Help Desk.

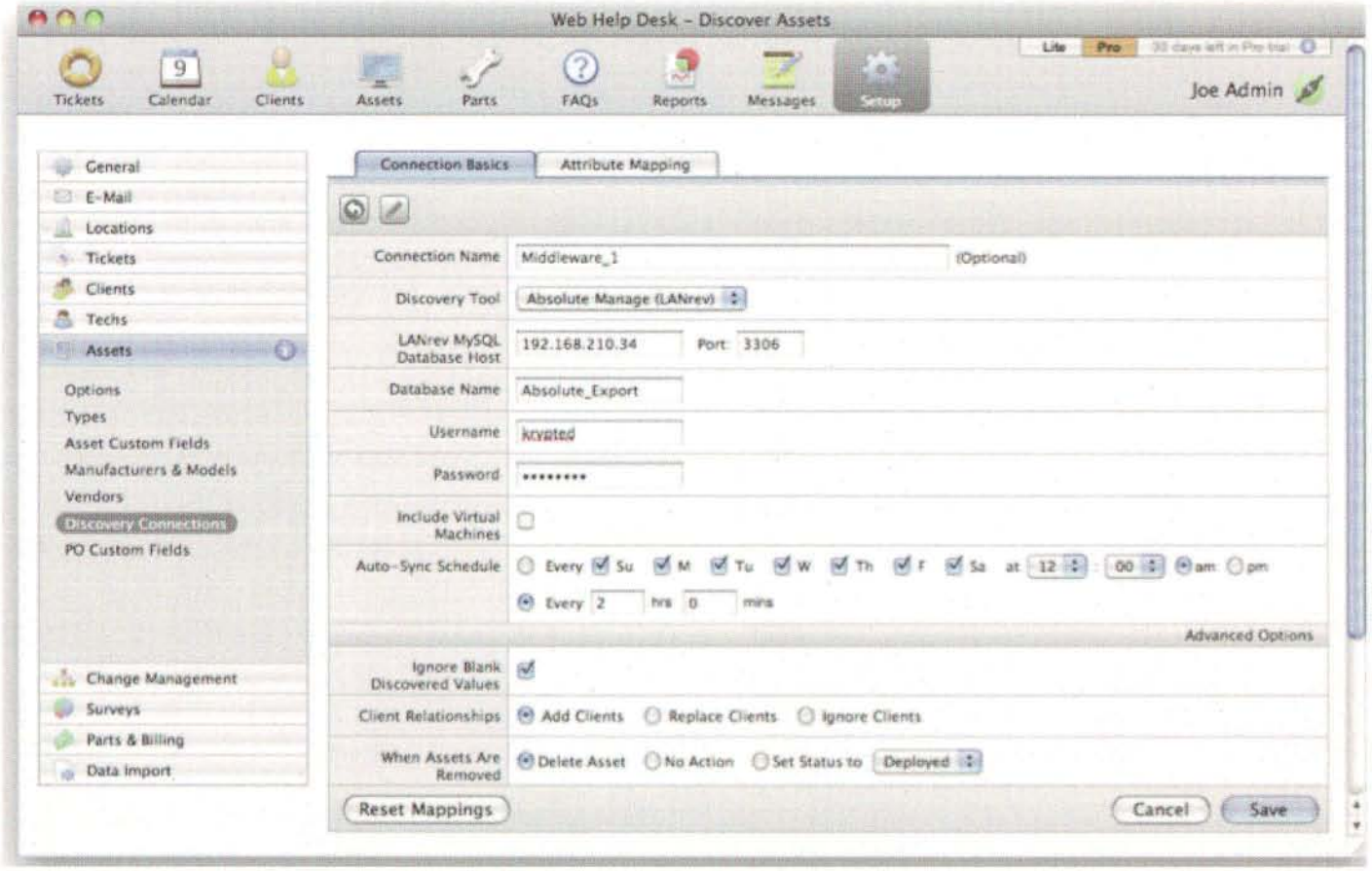


Figure 3 – Configuring Web Help Desk

The discovery tool should then propagate the machines from the intermediary database that was created at the next scheduled sync. In this example, we were able to interconnect two solutions without writing any scripts. We were able to do so because it is a common practice, where both vendors saw it fit to interlink their solutions. Web Help Desk is also able to import data from Apple Remote Desktop, JAMF's the Casper Suite and Microsoft's System Center Configuration Manager (SCCM). User accounts can also be imported into Web Help Desk using the LDAP Connections option listed under the Clients menu. Once data has been imported into Web Help Desk, administrators can then look at programmatically interfacing with Web Help Desk to create tickets.

Automated Ticket Management

Web Help Desk is trouble ticket management software that runs in Mac OS X, Windows Server and Linux environments. When using Web Help Desk, it is possible to bolt interoperability with databases and other solutions in by injecting information into URLs or leveraging emails that the server picks up and takes action on. This API functionality allows administrators to build middleware around the exposed features for such activities: logging into the database and creating tickets.

In the following examples, we will be using a server that we will call middleware.enterprisedesktopalliance.com. To simply bring up the login page, a script can call up the URL of the server, or when followed by/helpdesk/WebObjects/Helpdesk.woa/wa/login, other information can be injected into the URL. For example, this could be put into a script by using an open command along with the URL, a username (in this example cedge), a password (in this exampleSuperSecret):

```
open
http://middleware.enterprisedesktopalliance.com/helpdesk/WebO
bjects/Helpdesk.woa/wa/login?username=cedge&password=SuperSec
ret
```

Note: While we embedded the password in the above command for simplicities sake, more logic could easily be added in a script to call the password from keychain or another more secure location, which would be much more appropriate in a production environment.

A common use for something like the above would be a Mac OS X menu item for an organization that loads up a site for creating tickets. But it is important to note that authentication is required for each URL sent. Therefore, administrators scripting against Web Help Desk will want to assume tasks will be performed on behalf of a common user, an issue in some environments. In addition to simply authenticating users to the server, scripts can also be used to generate tickets in Web Help Desk.

To create tickets, the /helpdesk/WebObjects/-Helpdesk.woa/wa/createTicket will be appended to the base URL. Looking at the previous URL, the options that are provided begin with the question mark (?) and are separated by ampersands (&). The options allowed include the following:

User Settings:

email: End user's email address

last_name: End user's last name

first_name: End user's first name

user_name: End user's user name

password: End user's password

location: location of user (if none is provided then the default location of the user is used)

department: user's department name (if none is provided then the default department for the user is used)

Ticket Setettings:

problem_id: Ticket problem type id number (integer for PROBLEM_TYPE)

priority_id: Ticket priority id number (integer for PRIORITY_TYPE)

subject: Ticket subject

detail: Information to be placed into the ticket problem detail text field

id: customer_id number (if using the hosted Web Help Desk, an organization ID will also be required, otherwise an option)

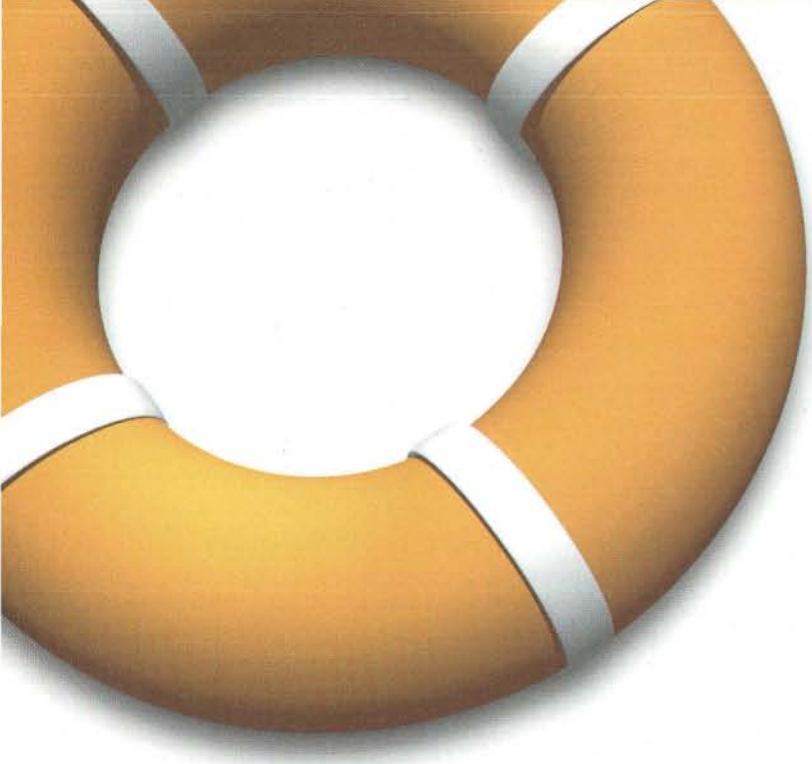
If a script attempts to create a ticket for a user who does not exist then the user will be created. In this case one should always try to provide an email address, although an account can be created even if there is no email address. This allows the automated creation of users using a looping script to, for example, create a ticket for them to provision their Web Help Desk accounts.

Creating tickets is one of the more important tasks though. To do so is straightforward. Simply use the same structure as the previously used command and make sure to use Unicode appropriate characters (spaces are not allowed). The following command would create a new ticket for Charles Edge, at my home, with an email address of enduser@enterprisedesktopalliance.com with a problem_id of 2 and a priority_id of 4:

```
open
http://mydomain.com/helpdesk/WebObjects/Helpdesk.woa/wa/creat
eTicket?problem_id=2&priority_id=4&subject=New%20Laptop&detai
l=My%20Windows%20Netbook%20is%20heavy&email=enduser@enterpris
edesktopalliance.com&first_name=Charles&last_name=Edge&locati
on=Home
```

We were again using the open command; however, the curl command could also be used. Keeping the entire transaction within a script and not actually opening a browser window is often times best when working with these sorts of tasks so as not to confuse users. Additionally, perl, python, Visual Basic, .Net and practically every other scripting environment is going to allow handling URLs in a far more graceful manner than using the open command (but it makes for a easy example).

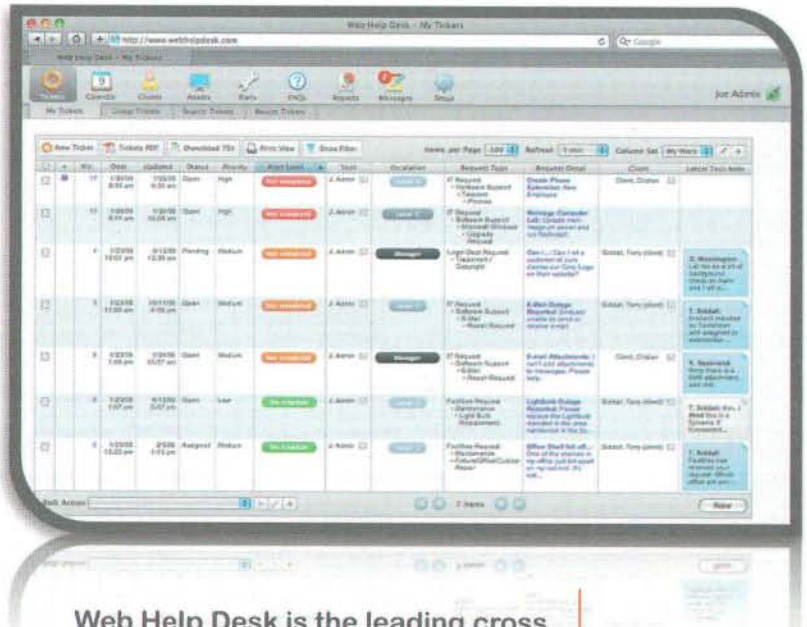
Once created, the client's ticket history page is returned, with a message acknowledging the new ticket. This message



Help Has Arrived.

Web Help Desk

Powerful Software for Service Management



Web Help Desk is the leading cross platform service management solution for support professionals seeking to simplify and control their increasingly complex service management environments.

Contact us today to learn more.
www.WebHelpDesk.com

Web Help Desk is a Proud Member of

Absolute Software · Centrify · ExtremeZ-IP · Web Help Desk · Enterprise Desktop ALLIANCE

MACTECH domains

Register

Get your .COM or any other domain name here!

FREE with every domain:

- FREE! Starter Web Page
- FREE! Getting Started Guide
- FREE! Complete Email
- FREE! Change of Registration
- FREE! Parked Page w/ Domain
- FREE! Domain Name Locking
- FREE! Status Alert
- FREE! Total DNS Control

Just visit
www.mactechdomains.com
to register for your domain today!

Starting at $1.99

when a non-domain name product is purchased. Limitations apply.

could be used to generate a response either to a script or to an application. If an error occurs the login page is returned, with a messaging describing the error. If the specified e-mail address cannot be found in the CLIENT table, an attempt is made to create a new client account (see notes below).

Conclusion

Most software packages used by Systems Administrators have an API of some sort. Those referenced in this article are interacting with Web Help Desk, Centrify, ExtremeZ-IP and Absolute Manage using different methods in order to showcase the different features of each. However, what we have done is really only the tip of the iceberg. There are no limits to what you can do when it comes to extending the functionality of software when there is a strong business case for doing so.

When costs are justified, many will look at the least expensive options for scripting. Expense can mean a variety of things. It can mean, expense in terms of CPU or RAM utilization, it can mean expense in terms of time required to program a given solution or it can mean expense in terms of actual cost. These three do no always line up accordingly. But, you will often find that coding something in bash is going to come with limitations, both in terms of scale and resource utilization faster than going more direct to the application, as is made possible by writing a tool in WMI or PowerShell. Having said this, WMI or PowerShell are going to have a far steeper learning curve.

Whichever you choose, take these examples and build on them. If features that you need aren't referenced, review the documentation links provided or contact the vendor and let them know what you would like to do and see if someone else has already done it. While we discussed throughout this article the fact that environments are often very different, it just so happens that rarely are any of us going to try something that hasn't been done before. If something has been done, then there is no reason to reinvent the wheel. Take that time you would be doing so and have a little fun. You deserve it (if only for having made it all the way through this article).

About The Authors

Charles Edge is the Director of Technology for 318 Inc, a national provider of IT Services with a focus on the Apple platform. Charles is also the author of a number of books on Mac Systems Administration, including the Enterprise Mac Administrator's Guide from Apress. You can reach him at cedge@318.com.

The members of the Enterprise Desktop Alliance are committed to making it easy to deploy, integrate and manage Macs in a Windows environment. The authors are technical experts within Centrify and Group Logic. info@enterprisedesktopalliance.com

Use the resources you already have to easily manage and secure

Macs in the Enterprise

The Designer
Needs her MacBook in order to work productively and cheerfully

The IT Director
Has to keep expenses and overhead in line while assuming responsibility for Macs

The CEO
Expects IT to support his Mac, which he uses along with his iPhone and iPad

They can all be happy

Everyone from the CEO to part-time contractors want to bring Macs into the workplace. But your Windows-centric IT department worries that they'll need additional hardware and training to efficiently manage and secure them.

With Centrify, you can use your existing Windows tools and infrastructure to centrally manage user accounts, control access across departments or geographies, and even use Group Policy to ensure compliance with password, screensaver lockout, file-sharing and other security measures.

Best of all, you can get started for free! With Centrify Express, in minutes you can have users logging into their Macs with their corporate Windows username and password.

Centrify®

Get Started for FREE!

www.centrify.com/express

Centrify is a Proud Member of

Absolute Software · Centrify · ExtremeZ-IP · Web Help Desk · Enterprise Desktop ALLIANCE

The SQLite DB and iOS

How to use SQLite on your iPhone and iPad

by Mihalis Tsoukalos

Introduction

This article will show you how to use the SQLite database from a UNIX shell and how to program it for iOS applications.

Because the programming interface to the SQLite database is written in C and Objective-C is a superset of C, it is easy to make use of SQLite into your iPhone projects.

After reading this article, you will be ready to understand the advantages of SQLite3 and programmatically use the SQLite3 database either from the UNIX command line or your iOS device.

At the time of writing this article the latest SQLite version is 3.7.3 and the latest Mac OS X version is 10.6.5.

Introduction to the SQLite DB

SQLite does not offer authentication or authorization. UNIX file permissions (using the *chmod* command) are used in order to determine the three SQLite-supported access levels: read/write access, read access and no access.

Also, SQLite is not suitable for very large datasets even though modern filesystems support files with sizes bigger than a terabyte.

Lastly, SQLite does not support replication–you can backup a database by simply copying the database file!

The advantages of SQLite are the following:

It has great performance.
It is reliable.
It is portable.
It is self-contained (the main reason that it was embedded for iOS).
It has a small runtime footprint – small size as well as small memory usage.
You do not need a GUI to use it.
You do not need to setup/start a server process to use SQLite.
Supports the query languages features of the SQL92 standard.

Using the SQLite DB from the Terminal

First of all, let me tell you the reason you need to learn how to use SQLite from the Mac OS X command line: you can easily create a database and copy it inside an iPhone Xcode project in order to use it from your iPhone application!

The simple database that you are going to create in this section is going to be used in the forthcoming sections of this article.

The following are the most important operations that you should know when using an SQLite database.

Creating a database

The following command will create a new database (by simply creating a new file) if the file does not already exist:

```
$ sqlite3 testSQLite.db
SQLite version 3.7.3
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
```

Deleting a database

In order to delete a database, you only have to delete the relevant file. In our case the following command will be enough:

```
$ rm testSQLite.db
```

Creating a table

After creating a database and running the *sqlite3* command, you will have to type the following in order to create a new table called *test*.

```
sqlite> create TABLE "test" ( "Name" TEXT, "Surname" TEXT);
```

The simple table called *test* has two fields called *Name* and *Surname*, both of them having the TEXT type.

Inserting data into a table

The following three commands insert three entries into our *test* table.

```
sqlite> INSERT INTO test ("Name", "Surname")
VALUES ("Mihalis", "Tsoukalos");
sqlite> INSERT INTO test ("Name", "Surname")
VALUES ("MacTech", "Magazine");
sqlite> INSERT INTO test ("Name", "Surname")
VALUES ("Edward", "Marczak");
```

Displaying the data of a table

The following command will display the data that the *test* table has.

Richard|Solo®

Backup Battery Solutions for iPhone/iPod/iPad

RichardSolo 1800 mAh

Backup Battery with laser beam, ultra-bright flashlight, wall charger, car charger, and charging cable.

RS001 For iPhone 2G/3G/3GS/4 and iPod
Now $19.95

RS008 With Cable for iPhone 2G/3G/3GS/4 and iPod
Now $29.95

Made for iPod iPhone

iPod is a trademark of Apple Inc., registered in the U.S. and other countries. iPhone is a trademark of Apple Inc.

Metal-cased backup battery with nylon hard case, one USB port!
RichardSolo 9000 mAh Universal Mobile Charger
VT900 **$69.95**

Charge anything with built-in USB port!
RichardSolo 4400 mAh Battery Case for iPad
TX440 **$79.95**

Details Online
RichardSolo.com

1700 mAh built-in battery, rubberized finish, for 100% all-day power!
RichardSolo Power Case for iPhone 4
AU001 **$59.95**

Bluetooth + quality hard keys, for iPad and all phones with BT!
Portfolio with Removable Bluetooth Keyboard
QM001 **$109.95**

Weekly GREAT DEAL

The **WEEKLY GREAT DEAL** offers incredible savings! Visit RichardSolo.com to sign up... it's free and fun.

FREE SHIPPING!
Enter the discount code ***techoff*** at checkout!

Order Now! RichardSolo.com

```
sqlite> select * from test;
Mihalis|Tsoukalos
MacTech|Magazine
Edward|Marczak
```

Please consult the Web Links and Bibliography section of this article for more resources about both SQLite3 and the SQL language.

Programming the SQLite DB Using Objective-C

First of all, forget the title of this section! The truth is that you will need some C (because SQLite's API is written in C) to program the SQLite DB.

The critical side effect of this fact is that you will not be able to use *NSString* objects (that are needed for iPhone programming) to pass data to the SQLite DB because C knows nothing about *NSString* objects!

So, you will have to wrap C code inside your Objective-C code that is required to "manually" convert your C strings (that are actually C *character pointers (char *)*) into Objective-C *NSString* objects and vice versa in order to exchange information between the two programming languages. In order to convert a *NSString* object into a C string, you need to use the *UTF8String* method of the *NSString* class and the *initWithUTF8String* method in order to do the reverse thing. The example code is the following:

```
#import <Foundation/Foundation.h>
#import <sqlite3.h>

// Programmer: Mihalis Tsoukalos
// Date: Monday 25 October 2010
//
// This is an example program of using
// SQLite using Objective-C
//
// Compile it with the following command:
//
// gcc -framework Foundation SQLiteExample.m -o
SQLiteExample -lsqlite3
//

int main (int argc, const char * argv[])
{
   NSAutoreleasePool * pool = [[NSAutoreleasePool alloc]
init];

    sqlite3 *database;
    sqlite3_stmt *statement;

    NSString *filePath = @"testSQLite.db";

    // A File Manager for file operations
    NSFileManager *fm = [NSFileManager defaultManager];


    // Check if the database file already exists
    BOOL exists = [fm fileExistsAtPath:filePath];

    if (exists)
        NSLog(@"Database file exists!");
    else
    {
        NSLog(@"Database file does not exist. Quiting...");
        return 1;
    }
```

```
    const char *cfilePath = [filePath
cStringUsingEncoding:NSUTF8StringEncoding];
    if ( sqlite3_open(cfilePath, &database) != SQLITE_OK )
    {
        NSLog(@"Cannot open database!");
        NSLog(@"Quiting...");
        return 2;
    }
    else
        NSLog(@"Database %@ opened!", filePath);

    // The cStatement has to be converted
    // into a sqlite3_stmt.
    const char *cStatement = "SELECT * FROM test";

    if (sqlite3_prepare(database, cStatement, -1, &statement,
NULL) != SQLITE_OK)
    {
        NSLog(@"Error preparing %s", cStatement);
        NSLog(@"Quiting...");
        return 3;
    }

    while (sqlite3_step(statement) == SQLITE_ROW)
    {
        // Read a C string
        const char *cName = (const char *)
sqlite3_column_text(statement, 0);
        // Convert it to NSString
        NSString *name = [[[NSString alloc]
initWithUTF8String:cName] autorelease];

        // Read a C string
        const char *cSurname = (const char *)
sqlite3_column_text(statement, 1);
        // Convert it to NSString
        NSString *surname = [[[NSString alloc]
initWithUTF8String:cSurname]
autorelease];

        // And now, lets display the data
        NSLog(@"name = %@ and surname = %@", name, surname);
    }

    // Clear the query results
    sqlite3_reset(statement);
    sqlite3_finalize(statement);

    // Now close the database
    sqlite3_close(database);

    [pool release];
    return 0;
}
```

As I say inside the objective-C code, in order to compile it you need to run the following command from the Terminal application:

```
$ gcc -framework Foundation SQLiteExample.m -o SQLiteExample
-lsqlite3
```

The *–lsqlite3* parameter tells the **gcc** compiler to link the sqlite3 library when compiling the Objective-C code.

The *–framework Foundation* parameter tells the **gcc** compiler that the code to be compiled is written in the Objective-C programming language and therefore needs to be linked with the Foundation Objective-C library.

So, if you run the executable (that will be called *SQLiteExample* because of the *–o* parameter) you will see the following (depending on the data that you put inside the *test* table):

```
2010-11-13 21:09:17.327 SQLiteExample[79781:903] Database file exists!
2010-11-13 21:09:17.331 SQLiteExample[79781:903] Database
testSQLite.db opened!
2010-11-13 21:09:17.331 SQLiteExample[79781:903] name = Mihalis and
surname = Tsoukalos
2010-11-13 21:09:17.332 SQLiteExample[79781:903] name = MacTech
and surname = Magazine
2010-11-13 21:09:17.332 SQLiteExample[79781:903] name = Edward
and surname = Marczak
```

The *sqlite3_finalize()* routine destroys a prepared statement created by a prior call to *sqlite3_prepare()*. Every prepared statement must be destroyed using a call to this routine in order to avoid memory leaks. If you do not execute the *sqlite_finalize()* routine you will get the "unable to close due to unfinalised statements" error.

Using the SQLite DB from the iPhone

The fist thing that you need to do is to manually create an SQLite database file using the Terminal application.

Then, you need to copy that file into the Resources of the application.

Please do not forget to save a copy of the database file if case you want to edit it later.

Last, you will need to add the `libsqlite3.dylib` file to your project. The `libsqlite3.dylib` file is a link that points to the latest version of the SQLite3 library. If you do not add the `libsqlite3.dylib` file then you will see an image with error messages similar to Figure 1 and the compilation process (actually the linking part of it) will fail.

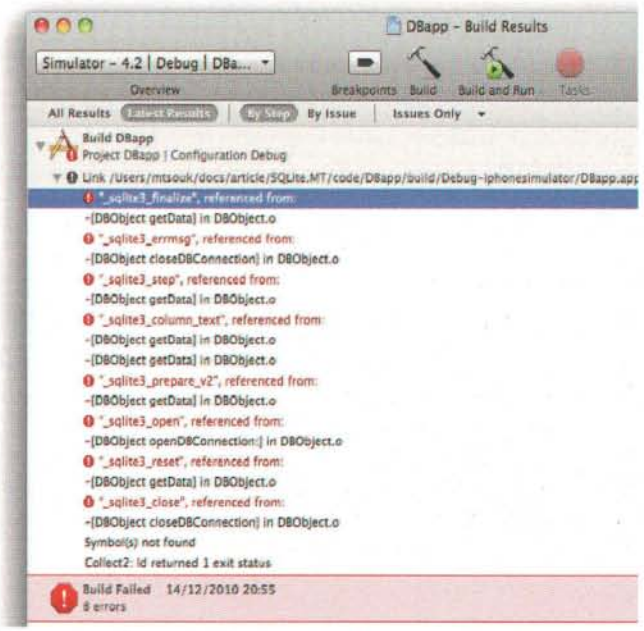


Figure 1: Compiling without linking to the SQLite library

The process to add a library is the same as adding a *Framework* and is as follows. In Xcode:

Select "*Frameworks*" from the "*Group & Files*" pane and right click on it.

From there, select "*Add*" and then "*Existing Frameworks...*"

From the really big list (figure 2), select *libsqlite3.dylib* (which is the same as selecting *libsqlite3.0.dylib*) and click add.

Build and Run your project!

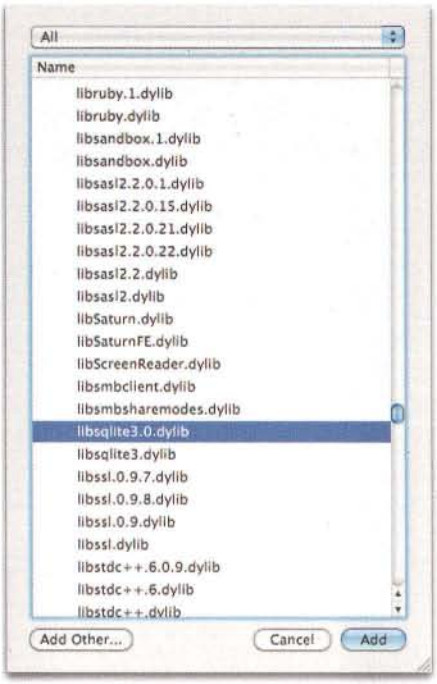


Figure 2: Adding the libsqlite3.dylib library

An iPhone application that uses SQLite

This is the interesting part of the article where we stop talking theoretically and start doing some practical things. You are going to make a complete iPhone application that uses SQLite. Xcode will be used for writing and compiling the application.

Also, if you want to run the application using your iPhone and not the iPhone Simulator, you will need to go to the iOS Dev Center and enroll to the iOS Developer Program. I am currently using iOS 4.2.1 on my iPhone 4.

The iPhone application that this article is going to program will display the contents of the *test* table (created in a previous section of this article). You can download all the classes from ftp://ftp.mactech.com.

First of all, create a new iPhone project from the existing Application templates. Choose "Navigation-based Application" as you can see in figure 3. The name of the application will be **DBapp**.

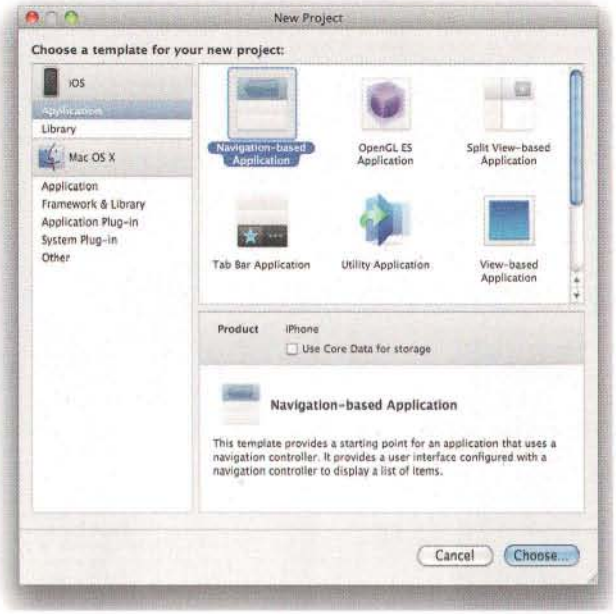


Figure 3: Creating a new Navigation-based Application

In order to be able to re-use some of the presented code in other projects, a separate class that deals with the SQLite database access will be introduced. The class will be called DBObject and will be implemented in two separate files, the DBObject.h file that contained the class declaration and the DBOject.m file that contains the class implementation that is a standard Objective-C practice.

In order to include the DBOject class in your project, you will have to go to "File"->"New File" and then select "Objective-C Class" from the "Cocoa Touch Class" group.

The DBObject.h file is straightforward: it has one variable that holds the database connection and three class methods that are going to be implemented in the DBOject.m file.

You should always check the for the *SQLITE_OK* return value to make sure that everything worked successfully. If there is an error, you can show it as in the following piece of code:

```
if (sqlite3_close(database) != SQLITE_OK)
    {
        NSLog(@"Error: failed to close database: '%s'.",
sqlite3_errmsg(database));
    }
    else
    {
        NSLog(@"Database connection successfully closed");
    }
```

The *init* function is automatically executed from the *RootViewControl.m* file using the following line of code:

```
DBObject *database = [[DBObject alloc] init];
```

The *init* function opens the database connection and then fetches the data using the next line of code:

```
self.data = [database getData];
```

You can also use the following way in order to convert an NSString into a C string:

```
const char *cfilePath = [dbFileSQ
cStringUsingEncoding:NSUTF8StringEncoding];
```

The *cStringUsingEncoding* method of the NSString class returns a C string using a given encoding.

You also need to add the SQLite database file—named *testSQLite.db*—to your Xcode project. You should right-click the Resources folder and select "Add" and then "Existing Files". You will then select the file and after than you will a figure similar to Figure 4. Please make sure that "Copy items into destination group's folder (if needed)" is selected.

Figure 4: Inserting the database file to your Xcode project

You will also need to create another class that will hold the data of each table row. I called it *person* and it has two members (*name* and *surname*). You will need to create its two files, the include file (*person.h*) and the implementation file (*person.m*).

If you want to create your own iOS application you will need to adjust the *person* class (or even change its name!) in order to contain the fields that match your needs and your database tables.

Also, you should alter the RootViewController.h and RootViewController.m files. Their full contents are the following:

```
//
//  RootViewController.h
//  DBapp
//
//  Created by Mihalis Tsoukalos on 15/11/2010.
//  Copyright 2010 yourCompany. All rights reserved.
//

#import <UIKit/UIKit.h>
#import "DBObject.h"
#import "person.h"

@interface RootViewController : UITableViewController
{
    NSMutableArray *data;
}

@property (retain, nonatomic) NSMutableArray *data;

@end
```

The contents of the *RootViewControl.h* file are simplistic whereas the altered *RootViewControl.m* file is more complicated. Nevertheless note that most of its code is automatically created by Xcode. The added code is in **bold** typeface.

```
//
//  RootViewController.m
//  DBapp
//
//  Created by Mihalis Tsoukalos on 15/11/2010.
//  Copyright 2010 yourCompany. All rights reserved.
//

#import "RootViewController.h"


@implementation RootViewController

// This completes the @property command
// from the RootViewController.h file
@synthesize data;

#pragma mark -
#pragma mark View lifecycle

- (void)viewDidLoad
{
    [super viewDidLoad];

    // Uncomment the following line to display an Edit
button in the navigation bar for this view controller.
    // self.navigationItem.rightBarButtonItem =
self.editButtonItem;

    DBObject *database = [[DBObject alloc] init];
    self.data = [database getData];

    [database closeDBConnection];

    [database release];
}


#pragma mark -
#pragma mark Table view data source

// Customize the number of sections in the table view.
- (NSInteger)numberOfSectionsInTableView:(UITableView
*)tableView {
    return 1;
}


// Customize the number of rows in the table view.
- (NSInteger)tableView:(UITableView *)tableView
numberOfRowsInSection:(NSInteger)section
{
    return [self.data count];
}


// Customize the appearance of table view cells.
- (UITableViewCell *)tableView:(UITableView *)tableView
cellForRowAtIndexPath:(NSIndexPath *)indexPath {
```

```
    static NSString *CellIdentifier = @"Cell";

    UITableViewCell *cell = [tableView
dequeueReusableCellWithIdentifier:CellIdentifier];
    if (cell == nil) {
        cell = [[[UITableViewCell alloc]
initWithStyle:UITableViewCellStyleDefault
reuseIdentifier:CellIdentifier] autorelease];
    }

    // Configure the cell.
    person* myPerson = [self.data objectAtIndex:[indexPath row]];
    cell.textLabel.text = myPerson.surname;

    return cell;
}

#pragma mark -
#pragma mark Table view delegate

- (void)tableView:(UITableView *)tableView
didSelectRowAtIndexPath:(NSIndexPath *)indexPath {

    /*
     <#DetailViewController#> *detailViewController =
[[<#DetailViewController#> alloc] initWithNibName:@"<#Nib
name#>" bundle:nil];
     // ...
     // Pass the selected object to the new view
controller.
    [self.navigationController
pushViewController:detailViewController animated:YES];
    [detailViewController release];
    */
}

#pragma mark -
#pragma mark Memory management

- (void)didReceiveMemoryWarning {
    // Releases the view if it doesn't have a superview.
    [super didReceiveMemoryWarning];

    // Relinquish ownership any cached data, images, etc
that aren't in use.
}

- (void)viewDidUnload {
    // Relinquish ownership of anything that can be
recreated in viewDidLoad or on demand.
    // For example: self.myOutlet = nil;
}


- (void)dealloc
{
    [super dealloc];
}

@end
```

If you have no typos, then by compiling and running the application you will see Figure 5.

Also, the application only displays the *surname* column from the *test* table. This is done by using the following code:

```
cell.textLabel.text = myPerson.surname;
```

Figure 5: The DBApp is running!

You can further change the look of the application by customizing the *UITableView*.

Summary

The simple application that was programmed in this article is a complete example of an iPhone application that uses the SQLite3 database.

Its output is minimal but the presented code fully shows the use of SQLite3 using Objective-C and Xcode.

SQLite3 is very efficient, uses SQL which is the standard language for creating, querying and changing databases and is embedded in every iPhone and iPad. What else can you ask?

Web Links and Bibliography

SQLite site: http://www.sqlite.org/
Using SQLite: http://oreilly.com/catalog/9780596521196/
iOS Dev Center:
http://developer.apple.com/devcenter/ios/index.action
Using SQLite by Jay A Kreibich, O'Reilly, 2010
SQL Cookbook by Anthony Molinaro, O'Reilly, 2006
Learning SQL by Alain Beaulieu, O'Reilly, 2009
SQLite Manager plug-in for Firefox,
http://code.google.com/p/sqlite-manager/
iPhone Programming: The Big Nerd Ranch Guide by Joe Conway & Aaron Hillegass, Big Nerd Ranch, 2010

About The Author

Mihalis Tsoukalos enjoys digital photography, writing articles and programming his iPhone 4 and iPad. He is the author of* Programming Dashboard Widgets, *an eBook. You can reach him at tsoukalos@sch.gr.

/n software

www.nsoftware.com >

The Best-Selling Internet Communications, Security, and E-Business Components, Now Available On:

Mac OS X and iPhone!

COCOA FRAMEWORKS FOR INTERNET COMMUNICATIONS

IP*Works! eliminates the complexity of Internet development providing easy-to-use, programmable components that facilitate tasks such as sending E-mail, transferring files, managing networks, browsing the web, consuming Web Services, etc.

Internet Communications

- **IP*Works! - [Core Framework]**

A comprehensive framework for Internet development. The core building block for most /n software products.

Components - *IPMonitor, MX, REST, NetCode, RSS, NNTP, SMPP, POP, Rexec, Rshell, Syslog, SMTP, WebDav, SOAP, XMPP, Telnet, Ping, TFTP, FilerMailer, UDPPort, HTMLMailer, WebForm, NetClock, WebUpload, RCP, Whois, FTP, XMLp, HTTP, SNPP, IMAP, MIME, IPInfo, IPDaemon, IPPort, NetDial, LDAP, ICMPPort, MCast, TraceRoute*

Network Management

- **IP*Works! Secure SNMP**

A comprehensive toolkit for building Secure SNMP-based agent and manager applications including advanced SNMPv3 security features, trap handling, and ASN-1 MIB compilation.

Components - *SNMPAgent, SNMPMgr, SNMPTrapMgr, MibBrowser*

File & Streaming Compression

- **IP*Works! Zip**

Suite of easy-to-use, fast, effective components for compression and decompression with advanced features including self-extracting archives, industrial strength AES encryption, and Zip64 archives.

Components - *GZip, Tar, Jar, Zip, ZipStream*

Secure Connectivity & E-mail Confidentiality

- **IP*Works! SSL**

SSL-enabled versions of the components in the core IP*Works! package.

Components - *WebFormS, LDAPS, WebUploadS, RESTS, NNTPS, IPDaemonS, POPS, FileMailerS, SOAPS, HTMLMailerS, TelnetS, HTTPS, FTPS, IMAPS, SMTPS, IPPortS, CertMgr, RSSS, SMPPS, WebDavS, XMPPS*

- **IP*Works! S/MIME**

Components for E-mail and file confidentiality, authentication, and non-repudiation through encryption and digital signatures. Implements the S/MIME standard for digital security.

Components - *CertMgr, SNNTP, SPOP, SIMAP, SSMTP, SMIME, SFileMailer*

- **IP*Works! SSH**

Secure Shell (SSH) enabled client communications components supporting strong encryption and advanced cryptography. A highly-evolved code base, enhanced with enterprise features like IPv6 addressing and 64-bit support.

Components - *SFTP, SShell, SExec, SSHClient, SSHTunnel*, CertMgr

COCOA FRAMEWORKS FOR INTERNET BUSINESS

Built using the same technologies as our award-winning IP*Works! product line, these packages offer native components for Credit Card Processing, Online Payments, E-Banking, Shipping & Tracking, and more!

- **QuickBooks Integrator**
- **E-Payment Integrator**
- **E-Banking Integrator**
- **TSYS Integrator**
- **Paymentech Integrator**
- **FDMS Integrator**
- **FedEx Integrator**
- **USP Integrator**
- **USPS Integrator**
- **PayPal Integrator**
- **Amazon Integrator**
- **SharePoint Integrator**

GET THEM ALL THROUGH OUR COMPREHENSIVE RED CARPET SUBSCRIPTION, $1,499 PER DEVELOPER, ROYALTY-FREE. 20% DISCOUNT FOR MACTECH SUBSCRIBERS.

Adaptive Laptop Management with crankd

by Gary Larizza

Supporting laptops in a managed environment is tricky (and doubly so if you allow them to be taken off your corporate network). While you can be reasonably assured that your desktops will remain on and connected during the workday, it's not uncommon for laptops to go to sleep, change wireless access points, and even change between an Ethernet or AirPort connection several times during the day. It's important to have a tool that can "tweak" certain settings in response to these changes. This is where crankd comes in.

Crankd is a cool utility that's part of the Pymacadmin (http://code.google.com/p/pymacadmin/) suite of tools, co-authored by Chris Adams and Nigel Kersten. Specifically, crankd is a Python daemon that lets you trigger shell scripts, or execute Python methods, based upon state changes in SystemConfiguration, NSWorkspace and FSEvents. It's easier to see how crankd can help you by providing a couple of scenarios:

Use Cases

1. Your laptops, like all of the other machines in your organization, are bound to your corporate LDAP servers. When they're on-network, they will query the LDAP servers for things like authentication information. Unless your corporate LDAP directory is accessible outside your corporate network, your laptops may exhibit the "spinning wheel of death" when they attempt to contact a suddenly-unreachable LDAP directory at the neighborhood Starbucks. A solution to this is to remove the LDAP servers from your Search (and Contacts) path whenever the laptop is taken off-network and add the LDAP servers when you come back on-network.
2. Perhaps you're using Puppet, Munki, Chef, StarDeploy, Filewave, Absolute Manage, Casper, or any other configuration management system that needs to contact a centralized server for configuration information. Usually these tools will have your machine contact their servers once an hour or so, but this can be a problem if the machine is constantly sleeping and waking (such as Laptop Cart Labs or demonstration machines). Plus, if you take your machine off-network, you don't want it trying to contact a server that might not be reachable from the outside world. It would be nice to have your laptop "phone home" when it establishes a network connection on your corporate network and skip this step when the laptop is taken outside your organization.
3. OS X allows you to set a preferred order for your network connections, but it would be nice to disable the AirPort when your laptop establishes an Ethernet connection on your corporate network.
4. Finally, maybe you have the need to perform an action whenever your laptop sleeps (or wakes), changes a network connection, mounts a volume, or runs a specific Application (whether it's located in the Applications directory or anywhere else on your machine).

All of these situations can be made trivial through the help of crankd.

How do I get it working?

Crankd is a daemon, so it's running in the background while you work. It uses an XML plist file that lists the scripts (or Python methods) that crankd should execute in response to specific state changes (like a network connection going up or down or a volume being mounted). Since it's a small Python library, the files aren't huge and the entire finished installation is around 100 Kb (or larger with your custom code/scripts). Let's download crankd and experiment with its settings:

1. **Download the Pymacadmin source.** You can do this through Google Code or Github - I recommend the Github method as it seems to currently be a newer version. You have two choices for downloading the source code: you can either navigate to http://github.com/acdha/pymacadmin, click the Downloads button, download either the .tar.gz or the .zip version of the source code and double-click on the file to expand it (which should open a folder named *acdha-pymacadmin-<combination of 7 numbers and letters>*), or you could use git to check out a version of the Pymacadmin repository with the `git checkout https://github.com/acdha/pymacadmin.git pymacadmin` command (Do note that git needs to be installed on your computer as it doesn't come natively

with OS X. There is a Mac installer that can be downloaded from http://code.google.com/p/git-osx-installer/).

2. **Install crankd.** Upon opening the pymacadmin folder, you should see a series of folders, readme files, and an `install-crankd.sh` installation script. Let's open Terminal.app and navigate to the pymacadmin folder that we expanded on our desktop (you can type `cd` into Terminal.app and then drag and drop the folder into the Terminal window. Hit the Return button on your keyboard to change to the directory). The `install-crankd.sh` script is executable, so run it by typing `sudo ./install-crankd.sh` into the Terminal window and hitting Return. Enter your password when it prompts you.
3. **Setup a plist file for crankd.** Crankd uses an XML configuration plist to determine which state changes to monitor and which scripts or Python methods to execute in response to these state changes. If you've never worked with crankd before, it's best to let it create a sample configuration plist for you. If you don't specify a configuration plist with the `–config` argument, or you don't have a com.googlecode.pymacadmin.crankd.plist file in your /Users/<username>/Library/Preferences folder, crankd will automatically create this sample plist for you. Let's do that by typing `sudo /usr/local/sbin/crankd.py` into Terminal and hitting the Return button. Take a look at the sample configuration plist file it will create (located in your /Users/<username>/Library/Preferences folder) :

com.googlecode.pymacadmin.crankd.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST
1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>NSWorkspace</key>
    <dict>
        <key>NSWorkspaceDidMountNotification</key>
        <dict>
            <key>command</key>
            <string>/bin/echo "A new volume was
mounted!"</string>
        </dict>
        <key>NSWorkspaceDidWakeNotification</key>
        <dict>
            <key>command</key>
            <string>/bin/echo "The system woke from
sleep!"</string>
        </dict>
        <key>NSWorkspaceWillSleepNotification</key>
        <dict>
            <key>command</key>
            <string>/bin/echo "The system is about to
sleep!"</string>
        </dict>
    </dict>
    <key>SystemConfiguration</key>
    <dict>
        <key>State:/Network/Global/IPv4</key>
        <dict>
            <key>command</key>
            <string>/bin/echo "Global IPv4 config
changed"</string>
        </dict>
    </dict>
</dict>
</plist>
```

This XML file has two main keys - one for NSWorkspace events (such as mounting volumes and sleeping/waking your laptop), and one for SystemConfiguration events (such as network state changes) followed by a key for the specific event that we're monitoring, a key specifying whether we'll be executing a command or a Python method in response to this event, and a string (or an array of strings, as we'll see later) specifying the actual command that's to be executed. For all of the events in the sample plist, we're going to be echoing a message to the console.

4. **Start crankd**. Once crankd has been installed and your configuration plist file is setup, you're ready to let crankd monitor for state changes. Let's start crankd with the sample plist that was created in the previous step by executing `sudo /usr/local/sbin/crankd.py –config=/Users/<username>/Library/Preferences/com.googlecode.pymacadmin.crankd.plist` in Terminal. Remember to substitute your username for <username> in that command (if you don't know your username, you can type `whoami` into Terminal and hit the Return button). If everything was executed correctly, you should see the following lines displayed in Terminal:

```
    Module directory /Users/<username>/Library/Application
Support/crankd does not exist: Python handlers will need to
use absolute pathnames
    INFO: Loading configuration from
/Users/<username>/Library/Preferences/com.googlecode.pymaca
dmin.crankd.plist
    INFO: Listening for these NSWorkspace notifications:
NSWorkspaceWillSleepNotification,
NSWorkspaceDidWakeNotification,
NSWorkspaceDidMountNotification
    INFO: Listening for these SystemConfiguration events:
State:/Network/Global/IPv4
```

It might look like Terminal isn't doing anything, but in actuality crankd is listening for changes. You can make crankd come to life by either connecting to (or disconnecting from) an AirPort network, sleeping/waking your machine, or mounting a volume (by inserting a USB memory stick, for example). Performing any of these actions will cause crankd to echo messages to your Terminal window. Here's the message I received when I disconnected from an AirPort network:

```
    INFO: SystemConfiguration: State:/Network/Global/IPv4:
executing /bin/echo "Global IPv4 config changed"
    Global IPv4 config changed
```

To quit this sample configuration of crankd, simply hold down the control button on your keyboard and press the "c" key. Congratulations, crankd is now up and running!

A more complex example

Let's look at a scenario that is close to my heart. I run Puppet (http://www.puppetlabs.com/puppet) to manage all of

our district's server, desktop, and laptop machines, and I would like to call Puppet anytime a laptop establishes a network connection inside our corporate network. This example will show how a Python method can be executed in response to a network state change. There are 3 main parts to my solution:

A crankd plist file that specifies which state changes we're monitoring and which Python methods to execute in response to those state changes.

Python code to execute in response to the network state change.

A launchd plist that will keep crankd running.

The crankd Plist File

We need to generate a plist file to identify which state change we're monitoring and exactly what action we should take when this state change occurs. I would recommend monitoring the State:/Network/Global/IPv4 System Configuration state – this covers any IPv4 interface on your system. You COULD monitor each BSD interface separately (with the State:/Network/Interface/en0/IPv4 and State:/Network/Interface/en0/IPv4 keys), but you would then need logic to determine which BSD interface (such as en0 and en1) corresponds to each network interface (such as Ethernet or AirPort). The entire plist file should look like this:

crankd-config.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>SystemConfiguration</key>
    <dict>

      <key>State:/Network/Global/IPv4</key>
       <dict>
         <key>method</key>
            <array>
                <string>CrankTools</string>
                <string>onNetworkChange</string>
            </array>
       </dict>

    </dict>
</dict>
</plist>
```

The main difference between this plist file and the automatically-generated crankd plist file is that we're not executing a script in response to the State:/Network/Global/IPv4 state change. Instead, we're using the `method` key which tells crankd that we're going to be executing a Python method. The next two string values are the specific Python class and method that should be executed in response to this state change. Finally, let's name this file crankd-config.plist and place it into the /Library/Preferences folder (NOT the /Users/<username/Library/Preferences folder).

The CrankTools Python Class

The CrankTools class will be where all the Python magic happens. This class will contain code to check if the laptop is on or off-network, code to call Puppet, and the specific method that will be called from crankd. It's **VERY IMPORTANT** that the name of the file matches the classname that was used in the crankd-config.plist file (this must match down to spelling and case). Because we specified *CrankTools* in the crankd-config.plist, the file must be called CrankTools.py. The method name is equally as important as the class name. Make sure that an `onNetworkChange` method exists in your `CrankTools` class (and that its spelling and case match with the crankd-config.plist file). When your Python class file is complete, make sure to save this file into the /Library/Application Support/crankd/ folder.

The first method, called `onNetworkChange`, will either call the `executePuppet` method if it's determined that we're on our corporate network (via the `onTheNetwork` method), or do nothing (if we're off-network). You can add complexity as you become more comfortable, but for now we're keeping it simple to demonstrate the process.

The `onTheNetwork` method is going to be specific for every environment. Only you will know the best way to determine whether you're on your corporate network. For this demonstration, I'm going to use the `/usr/sbin/scutil –r <hostname>` command to try and access a machine that's only accessible when I'm on my corporate network.

Finally, the `executePuppet` method is going to call a Puppet wrapper-script (/usr/bin/puppetd.rb) because Puppet has its own complexities and checks.

Here is what my CrankTools.py file looks like:

CrankTools.py

```
__author__ = 'Gary Larizza (gary@huronhs.com)'
__version__ = '0.1'

import syslog
import subprocess

syslog.openlog("CrankD")
_PUPPETD = '/usr/bin/puppetd.rb'
_SCUTIL = '/usr/sbin/scutil -r odm.huronhs.com'

class CrankTools():

    def onNetworkChange(self, *args, **kwargs):
        """Triggered when a "State:/Network/Global/IPv4" change
occurs.
        """

        if self.onTheNetwork():
            self.executePuppet()

    def executePuppet(self):
        """Simple utility function that calls puppet via subprocess.
        The _PUPPETD variable is set globally and corresponds to
        my puppet wrapper script.
        ---
        Arguments: None
        Returns: Nothing
        """
        syslog.syslog(syslog.LOG_ALERT, "Performing a Puppet
```

```
Run.")
        command = [_PUPPETD]
        task = subprocess.Popen(command,
        stdout=subprocess.PIPE, stderr=subprocess.PIPE)
        task.communicate()

    def onTheNetwork(self):
        """This function will check to see if we are on our
corporate network.
        —
        Arguments: None
        Returns:
            Either True or a blank string depending on if we locate
the desired host.
        """

        command = subprocess.Popen(_SCUTIL, shell=True,
        stdout=subprocess.PIPE,)
        netcheck =
command.communicate()[0].rstrip().split(",")

        for status in netcheck:
            if status == 'Reachable':
                syslog.syslog(syslog.LOG_ALERT, "We are on-
network.")
                return "true"
        syslog.syslog(syslog.LOG_ALERT, "We are off-
network.")
        return ""
```

A Launchd Plist

To start crankd whenever the computer is turned on (and to run it as the root user), we're going to be using a launchd plist file that's put into /Library/LaunchDaemons. Launchd plists are fairly easy to setup – take a look at this sample file:

com.googlecode.crankd.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>KeepAlive</key>
    <true/>
    <key>Label</key>
    <string>com.googlecode.crankd.plist</string>
    <key>ProgramArguments</key>
    <array>
        <string>/usr/local/sbin/crankd.py</string>
        <string>-config=/Library/Preferences/crankd-
config.plist</string>
    </array>
    <key>RunAtLoad</key>
    <true/>
</dict>
</plist>
```

If we deconstruct this plist file, we can see that it's running the command `/usr/local/sbin/crankd.py -config=/Library/Preferences/crankd-config.plist` whenever the computer is started. If the command is ever stopped (whether killed from the command line or via Activity Monitor), launchd will fire off another instance immediately (as long as this launchd plist is active). **MAKE SURE** that you've tested out your crankd setup before you start this launchd plist. If your crankd setup has any errors, launchd will continually restart crankd.py until you unload the launchd plist (you can load or unload launchd plists with the command `sudo /bin/launchctl load /Library/LaunchDaemons/<name of your launchd plist file>` - just substitute *unload* for *load* if you want to unload the plist).

Testing crankd

Now comes the time to test your crankd setup. There are MANY things that can go wrong (errors in your Python code, typos in your configuration plist files, logic errors, etc...), so it's very important to test crankd with EVERY possible state change you're monitoring. Just open a Terminal window and execute `sudo /usr/local/sbin/crankd.py -config=/Library/Preferences/crankd-config.plist` to test crankd from the command line. A **GREAT** feature of crankd.py is that it will reload itself any time a configuration file or monitored script/Python method is changed. This means that you don't have to kill and relaunch crankd.py every time you make a change to your CrankTools.py file (for instance). If you're running crankd.py from the command line, you can always use the print command in your Python code to debug sections of your code that might be causing errors. I highly recommend writing to the system log any time your Python code is making changes to the system. Not only will you be able to track changes centrally if you aggregate your log data, but you'll be able to trace back logic errors in your system logs.

Where to go for help

Crankd is a part of the Pymacadmin suite of system tools. Pymacadmin maintains its own google group at http://groups.google.com/group/pymacadmin - so feel free to post questions for the good of the order. My personal blog (http://glarizza.posterous.com) has a couple of entries on crankd, and you're welcome to check my Puppet repository on Github for the exact crankd code that I use on my machines (https://github.com/huronschools/Huron-City-Schools-Puppet-Repository/tree/master/files/crankd). Good luck and get cranking!

About The Author

Gary Larizza is the director of technology for a K-12 school district in northern Ohio. He currently maintains his Mac infrastructure through the use of Puppet and enjoys documenting his experiences in the hope that it will help others. All of his posts can be found at http://glarizza.posterous.com and you can usually find him complaining about something on Twitter at @glarizza.

VMware View Client for iPad

Windows on iOS—to go!

by Dennis Sellers

VMware's VMware View Client for iPad (http://www.vmware.com/products/view) enables users of the Apple tablet to access their virtual Windows desktops, applications and data from just about anywhere. It's available for free at the Apple App Store and is optimized for the iPad's high-resolution Multi-Touch display.

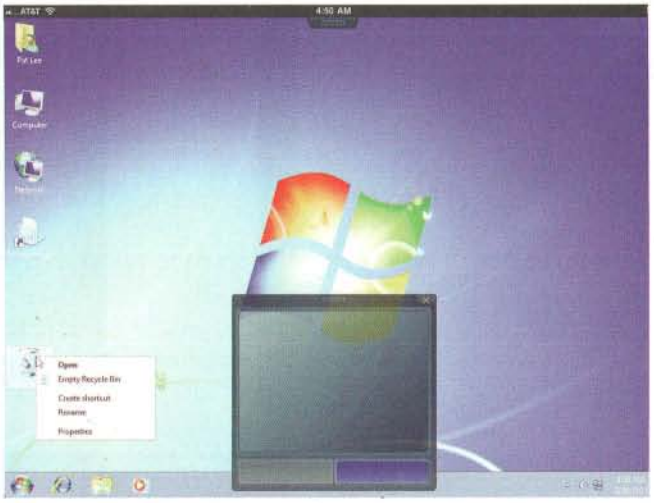
VMware View Client for iPad makes it easy to access your Windows virtual desktop from your iPad on the Local Area Network (LAN) or across a Wide Area Network (WAN). Since desktops are tied to users and not devices, desktops "follow" the user from device to device.

The new VMware View Client is the first iPad app to deliver Windows-based virtual desktops while taking advantage of the PC-over-IP (PCoIP) display protocol. On LAN or WAN connections, VMware View with PCoIP delivers a high performance desktop experience, even over high latency and low-bandwidth connections.

Custom gestures on the new VMware View Client enable navigation around the virtual desktop by taking advantage of iPad's Multi-Touch display. An on-screen track pad lets users leverage a more traditional mouse interface with the iPad's keyboard for text input.

VMware View offers Security Server support for PCoIP allows for a secure remote connection and authentication to a user's Windows desktop over WiFi or 3G neworks. The ability to select and connect to a list of recently connected desktops simplifies reconnection.

Support for the iPad Keyboard Dock and Bluetooth keyboards lets you input text with a physical, rather than virtual, keyboard, when you wish. The iPad VGA connector allows you to connect to an external monitor. VMware View Client for iPad supports iOS 4.2 and iOS 4.3.



The Windows interface on the iPad

Serving families in Southern California, Children's Hospital Central California recently deployed VMware View – a Windows compatible, client desktop product that provides remote desktop capabilities to users using VMware's virtualization technology — to provide "Follow-me Desktops" that move from room-to-room with clinicians and staff as they treat their patients. The hospital has plans to deploy the VMware View Client for iPad.

"The iPad could fundamentally change the way our clinicians and staff approach their IT needs," says Kirk Larson, chief information officer, Children's Hospital Central California. "Now with VMware View Client for iPad, our caregivers can have the freedom to access a patient's electronic medical records anywhere in the hospital via an iPad on a secure VMware View desktop. This could not only improve patient care but may enable us to dramatically reduce costs and simplify device management."

VMware View includes, and is tightly integrated with, VMware vSphere, a virtualization platform. It's designed to allow users to simultaneously power on thousands of desktops without performance degradation, and extend business continuity and disaster recovery features from VMware vSphere to your desktops. You can use VMware View to, among other things, deliver desktops to remote and branch offices to accelerate provisioning while retaining control.

Desktop and application virtualization breaks the bonds between the operating system, applications, user data and hardware, eliminating the need to install or manage desktop environments on end-user devices. From a central location you can deliver, manage and update all of your Windows desktops and applications.

VMware ThinApp application virtualization separates applications from underlying operating systems. Applications packaged with VMware ThinApp can be run in the datacenter where they are accessible through a shortcut on the virtual desktop, reducing the size of the desktop image and minimizing storage needs.

Since VMware ThinApp isolates and virtualizes applications, multiple applications or multiple versions of the same applications can run on the virtual desktops

An Awesome Email Marketing Tool for MACTECH READERS

Benchmark Email is the standard in permission-based email marketing

- ✔ List Management
- ✔ Free Newsletter Archive
- ✔ Easy to Use Drag-n-Drop Email Editor
- ✔ Powerful Personalization
- ✔ Image Gallery
- ✔ Visual Graphs for Open & Clickthrough Tracking
- ✔ Spam & Spell Checkers
- ✔ Creative & Compelling HTML Templates
- ✔ Upload Your own Template
- ✔ Easy Video Integration

MACTECH users get a 10% Lifetime Discount with this promo code 123923

Most email marketing services charge more money for less product. Not us. Benchmark Email's sophisticated suite of email marketing features lets you grow your list, send campaigns, track your data and even take online polls for an affordable price.

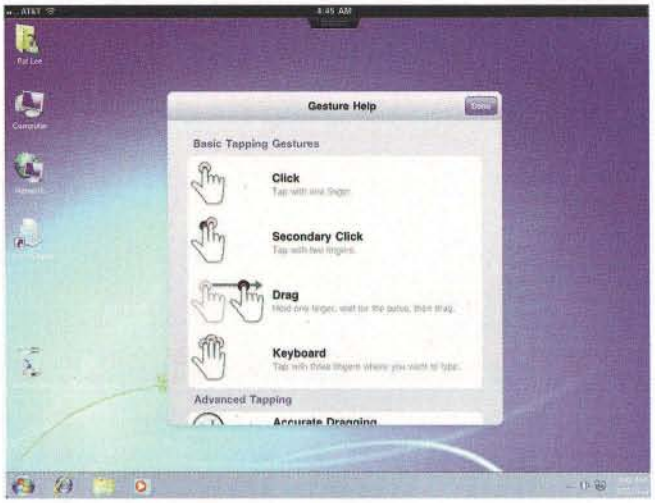
Plans starting at only $9.99 per month

Sign up for a FREE 30 Day Trial Today!

www.BenchmarkEmail.com

benchmark email

800.430.4095



Using Windows on the iPad presents some new challenges.

without conflict. Applications are assigned centrally through the View Manager, ensuring that all user desktops are up-to-date with the latest application versions.

VMware View Manager provides a single management tool to provision new desktops or groups of desktops, and an interface for setting desktop policies. Using a template, you can customize virtual pools of desktops and set policies, such as how many virtual machines can be in a pool, or logoff parameters.

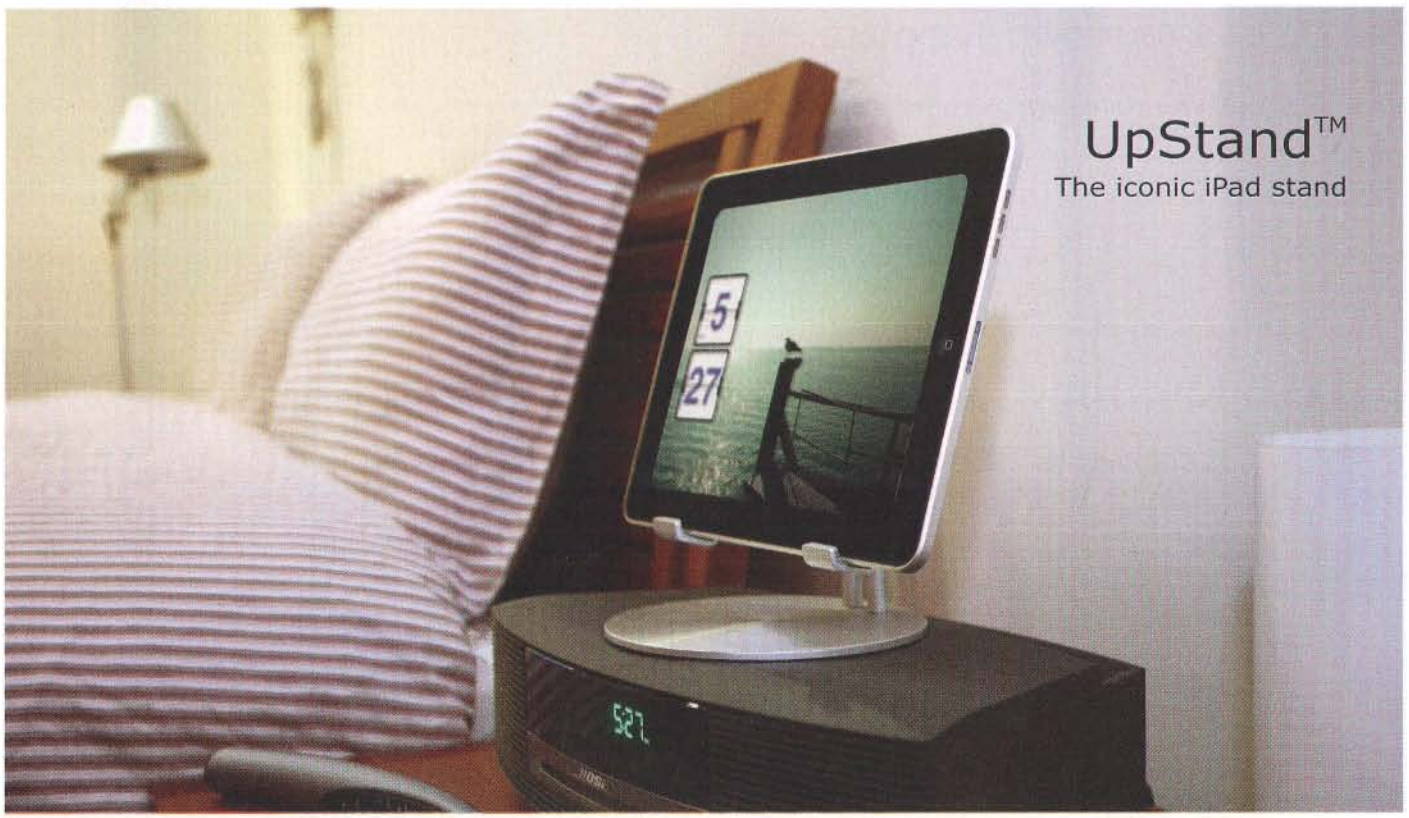
Based on the Linked Clone technology, VMware View Composer lets users create desktop images from a golden image. (A linked clone is a copy of a virtual machine that continues to share the virtual disks of its parents but basically runs of a snapshot, which preserves the exact state of the virtual machine when you create the clone). Updates implemented on the parent image can be pushed out to any number of virtual desktop quickly. With the core components of the desktop being managed separately, the process doesn't affect user settings, data or applications.

You can maintain control over data and intellectual property by keeping it secure in the datacenter. End users access their personalized desktop, complete with applications and data, securely from any location, at any time without jeopardizing corporate security policies. End-users outside of the corporate network can connect to their desktop securely through the VMware View Security Server.

Integration with vShield Endpoint enables offloaded and centralized anti-virus and anti-malware (AV) solutions. VMware View also supports integration with RSA SecureID for 2-factor authentication requirements.

About The Author

Dennis Sellers is a long time journalist. He started in the newspaper business, but has been in the online journalism business for the past 15 years. He's the editor/publisher of Macsimum News (http://www.macsimumnews.com)



Advertiser/Product Index

Ad Index by Company for: MacTech Mag 27.05 (May/2011)

Absolute Software....69
AlertSite....65
Aseopro....40
Audioengine....19
Benchmark Email....93
BITS Limited....32
Brad Sniderman....34
Centrify Corporation....79
Chlsoftware....37
Cocoatech....51
codefortytwo software....11
Dartware LLC....25
Digital Aurum....22
DR globalDirect, Inc. d/b/a eSellerate....47
DriveSavers Data Recovery....60
e3 Software....56
Elgato Systems GmbH....17
Enterprise Desktop Alliance....67
Faronics....39
Future Audio Design Innovation....31
Future Media Concepts....41
Gefen Inc.....46
Group Logic Inc.....73
Hansaworld....61
HostGator....49
iConnectivity....36
iDeveloper TV....63
IGC, Inc. / MaxEMail.com....54
Insider Software....41
IOGEAR....55
JAMF Software LLC....4
Just Mobile Ltd....94
LassoSoft Inc.....30
LC Technology International, Inc.....57
MacClinic/PC Clinic Sedona....26
MacsDesign Studio LLC....77
MacTech Conference....59
MacTech Domains....78
MacTech Boot Camp....23
MacTech DVD....33
Marketcircle Inc.....IFC
Microsoft....1
Mindraven....82
/n software inc.....87
Newsoft America....20
OlympicControls Corp.....43
Parallels Inc.....2-3
Pioneer....BC
Prosoft Engineering, Inc.....15
RD Ukraine LLC, dba Readdle....18
REAL Software, Inc.....29
RichardSolo....81
Ruckus Wireless....64
SAINT....21
SeedCode....22
SellYourMac....24
Small Dog Electronics....IBC
Smile....35
Sonnet Technologies, Inc.....14
Southern Stars....26
Stellar Information Systems Ltd.....50
Tethras....9
TransTech Systems....45
TrueShip....16
UsedCisco....13
VisiStat, Inc.....27
WebIS....42
ZAGG Inc.....50

Ad Index by Product for: MacTech Mag 27.05 (May/2011)

Accessories • IOGEAR....55
AppRadio • Pioneer....BC
Audio Speaker Systems • Audioengine....19
Audioglove • Future Audio Design Innovation....31
Benchmark Email • Benchmark Email....93
BookEndz • OlympicControls Corp.....43
Boot Camp • MacTech Magazine....23
Business Management Software • Hansaworld....61
Casper • JAMF Software LLC....4
Centrify Express • Centrify Corporation....79
Cisco Equipment • UsedCisco....13
CleanMyMac • Digital Aurum....22
CrashPlan • codefortytwo software....11
Data Recovery • DriveSavers Data Recovery....60
Data Rescue Center • Prosoft Engineering, Inc.....15
Daylite • Marketcircle Inc.....IFC
Deep Freeze • Faronics....39
Direct mail • e3 Software....56
DocMoto • Chlsoftware....37
Domain Registration • MacTech Domains....78
Drive Tool Box • Stellar Information Systems Ltd.....50
DVI to MiniDisplayPort • Gefen Inc.....46
Elgato • Elgato Systems GmbH....17
Enterprise Desktop Alliance Showcase • Enterprise Desktop Alliance....67
eSellerate • DR globalDirect, Inc. d/b/a eSellerate....47
ExtremeZ-IP • Group Logic Inc.....73
Font Agent • Insider Software....41
Hosting • HostGator....49
iConnectMIDI • iConnectivity....36
Identifying Solutions • TransTech Systems....45
iDeveloper TV • iDeveloper TV....63
InSight • AlertSite....65
InterMapper • Dartware LLC....25
IP*Works • /n software inc.....87
iPhone Accessories • RichardSolo....81
iPhone Apps • WebIS....42
IT Training • Future Media Concepts....41
Lasso • LassoSoft Inc.....30
Law Offices • Brad Sniderman....34
Localization Services • Tethras....9
MacTech Boot Camp • MacTech Magazine....23
MacTech Conference • MacTech Conference....59
MacTech DVD • MacTech Magazine....33
Manage • Absolute Software....69
maxemail.com • IGC, Inc. / MaxEMail.com....54
Microsoft Office • Microsoft....1
Parallels Desktop and Server • Parallels Inc.....2-3
Pathfinder • Cocoatech....51
PDF Expert • RD Ukraine LLC, dba Readdle....18
PDFPen • Smile....35
PHOTORECOVERY®/FILERECOVERY® • LC Technology International, Inc.....57
PrestoBizCard • Newsoft America....20
RackMac Mini • Sonnet Technologies, Inc.....14
REAL Studio Web Edition • REAL Software, Inc.....29
Retail Store • MacClinic/PC Clinic Sedona....26
Ruckus Wireless • Ruckus Wireless....64
Security Testing Tools • SAINT....21
Sell Your Mac • SellYourMac....24
SEO • Aseopro....40
SkySafari • Southern Stars....26
SmallDog.com • Small Dog Electronics....IBC
Smart Strips • BITS Limited....32
TrueShip Shipping System • TrueShip....16
UBB.threads • Mindraven....82
UpStand • Just Mobile Ltd....94
VisiStat • VisiStat, Inc.....27
Web Help Desk • MacsDesign Studio LLC....77
Zagg Skins • ZAGG Inc....50
Zulu iCal Server • SeedCode....22

THE MACTECH SPOTLIGHT

Justin Williams

SECOND GEAR

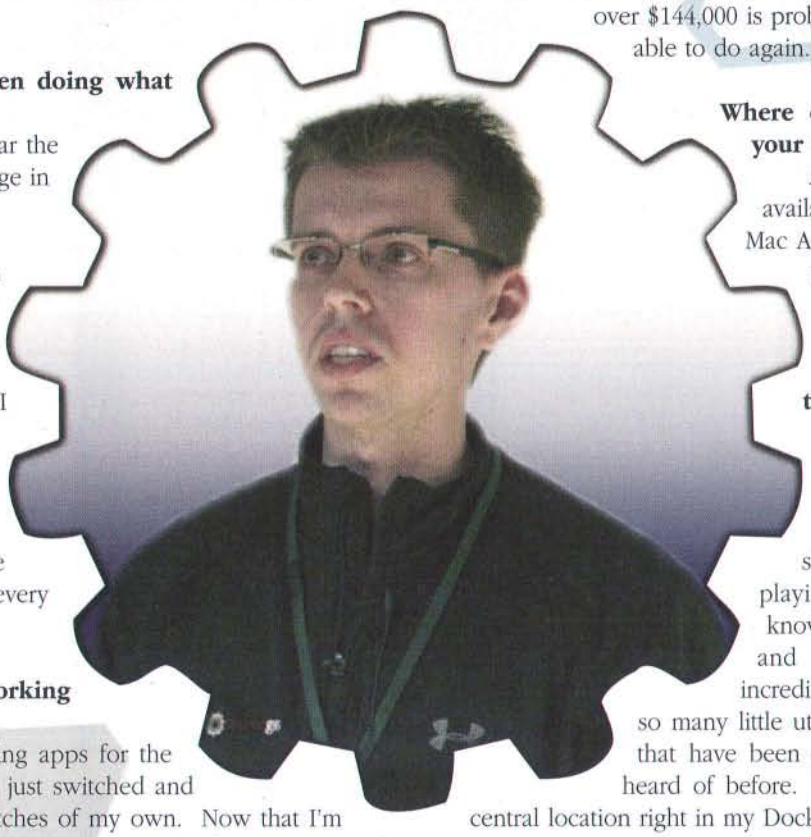
http://www.secondgearsoftware.com

What do you do?

Crew Chief, though I'm the only person on the crew so I guess I do everything.

How long have you been doing what you do?

I founded Second Gear the day after I graduated college in the spring of 2006.



Are you Mac-only, or a multi-platform person?

I use a Mac as my primary computer, but I switch between the iPhone, Windows Phone 7, Android and webOS for my smartphone. Women have shoes. I have phones for every occasion.

What attracts you to working on the Mac?

When I started building apps for the Mac it was because I had just switched and wanted to scratch a few itches of my own. Now that I'm several years in, all the improvements in the frameworks and tools make it hard for me to take other platforms seriously. Apple really understands that good tools are the key to a successful platform.

What's the coolest thing about the Mac?

I really like how most of the developers have taste. Great software on the Mac not only works great, but also looks great.

What is the advice you'd give to someone trying to get into this line of work today?

Just build something. Anything. Once you've polished it into a useful and good looking product, put it on the App Store and see if you can start building an audience.

What's the coolest tech thing you've done using OS X?

Building Elements, the first Dropbox powered text editor for iOS was pretty neat. It doesn't seem like much now as there are so many different choices out there, but being first to market has some clear advantages and was a real boost to the nerd ego.

Ever?

Organizing and coordinating Indie+Relief in January of 2010 was probably my greatest hit thus far. Indie+Relief (http://indierelief.com) was a 1 day charity event where hundreds of Mac and iOS developers donated their sales to Haitian Relief charities in the wake of the earthquake that hit the area earlier that month. It was a lot of work, but raising over $144,000 is probably something I'll never be able to do again.

Where can we see a sample of your work?

All of Second Gear's stuff is available on both the iOS and Mac App Stores. You can also see screenshots and other stuff at secondgearsoftware.com.

The next way I'm going to impact the Apple universe:

The Mac App Store is changing the way the desktop software industry is shaped. It's evening the playing field between widely known consumer app markets and the less obscure, but incredibly useful apps. I've found so many little utilities on the Mac App Store that have been out for years, but I'd never heard of before. Having all those apps in a central location right in my Dock is awesome.

Anything else we should know?

I tend to tweet quite a lot. Since I work alone at home, Twitter is my water cooler. I use the @secondgear account to about our products and interact with customers and am often mistaken for a teen popstar on my personal account @justin.

If you or someone you know belongs in the MacTech Spotlight, let us know! Send details to editorial@mactech.com

Mac shopping made easy.

Grab that to-do list, and prepare for some one-stop shopping at Smalldog.com!

Bundles simplify the buying process

Mac bundles (think Mac + RAM + AppleCare + external hard drive, etc.) not only include **everything you need,** but also **save you money.**

Visit » **Smalldog.com/specials**

Macs from under $500

We carry all **current Macs** as well as **used, refurbished and closeout models,** so there is a Mac for any budget.

Visit » **Smalldog.com/macs**

Free shipping over $200

It's true–we provide **free, same-day ground shipping** on every item over $200 every day.

Tax-free shopping

Purchases outside of Vermont are always shipped **tax-free.**

✓ 13" MacBook Pro + Chill Pill® mobile speakers

Small Dog Electronics

Always By Your Side

www.smalldog.com

800-511-MACS

Apple Specialist

Celebrating 15 Years • 3rd Largest Apple Specialist in New England • 5-Star Merchant Rating • Same-day shipping

Bundles

Macs

Free Shipping

Tax-Free

Pioneer

AppRadio™

A revolutionary way to use apps in the car

RETHINKING CONNECTIVITY

www.pioneerelectonics.com/appradio

PIONEER is a registered trademark of Pioneer Corporation. AppRadio is a trademark of Pioneer Corporation. ©2011 Pioneer Electronics (USA) Inc.
Limited apps are compatible with this model. Certain functions such as video playback and keyboard input are only available while parked.
For more information, see www.pioneerelectronics.com/appradio.